Technical University of Denmark

RELY 4. A Monte Carlo computer program for system reliability analysis

Kongsø, Hans Erik

Publication date:
1972

Document Version
Publisher's PDF, also known as Version of record

A. E. K. Risø

Risø - M - 1500

Risø - M - 1500

Title and author(s)

RELY 4. A Monte Carlo Computer Program for System Reliability Analysis

by

H.E. Kongsø

61 pages + tables + 11 illustrations

Date 6 June 1972

Department or group

Reactor Engineering Department

Group's own registration number(s)

Internal report no. 348

25-3-13

Abstract

A computer program for calculation of reliability and availability of systems is described.

The program is written in B6700 ALGOL.

The program can be used for reliability analysis of both mechanical, electric and electronic systems, also systems containing standby units.

The description consists of a summary, block diagrams and complete listings of the four versions of the program in addition to examples of input and output.

There are four versions of the program with different distribution functions for the times to failure and repairtimes. Two of the versions use variance reduction - importance sampling - and the others, direct simulation.

The basic theories and concepts are outlined in an appendix.

Copies to

Abstract to

Available on request from the Library of the Danish Atomic Energy Commission (Atomenergikommissionens Bibliotek), Risø, Roskilde, Denmark.
Telephone: (03) 35 51 01, ext. 334, telex: 5072.

F1 25-204

ISBN 87 550 0147 5

CONTENT

Side

1. INTRODUCTION 1

2. DESCRIPTION 2

2.1 General 2

2.2 Flow charts 3

2.3 Program listing 3

3. INPUT AND OUTPUT 31

4. EXPERIENCE 47

4.1 General 47

4.2 Version 1 47

4.3 Version 2 47

4.4 Version 3 48

4.5 Version 4 48

4.6 Future work 48

APPENDIX 49

A1 Definitions of reliability and availability 49

A2 Generation of random periods by the Monte Carlo technique 49

A3 Direct simulation 51

A4 Importance sampling 52

A4.1 Weighting factor for units without standby . 53

A4.2 Weighting factor for standby units 54

A4.3 Weighting factor for time independent failures 56

A4.4 Weighting factor for units with gamma- or exponentially distributed times to repair (version 1)................................ 56

A4.5 Weighting factor for units with a single standby unit and times to repair with a gamma- or exponential distribution (program version 1) 57

A4.6 Standard deviation of the result in version 1 and 3 58

A5 Calculation of the effective repair rate 59

ACKNOWLEDGEMENT 60

REFERENCES 61

FIGURES 62

1. INTRODUCTION

The Monte Carlo method is generally accepted as a very powerfull tool for calculation of reliability of systems, due to its versatility and - in particular for large complicated systems - relative short computation time if used in connection with a technique for reduction of the variance of the result, like importance sampling.

A litterature survey was carried out in order to find a description of an appropriate Monte Carlo program. Only one program of this type was found (ref.7). The name of this is SAFTE and it was developed by Holmes and Narver, Inc., Los Angelos. However, the SAFTE program as it appears in ref. 7 is not fully documented and is not able to account for standby-units. In the meantime two other Monte Carlo computer programs have been developed: The REMO code at Ispra, and the FBRR in Germany. The REMO code is able to account for standby-units and the FBRR is an implementation of SAFTE; none of these codes have yet been published.

This report contains a description of a Monte Carlo computer program for reliability calculation, developed at the reactor engineering department. The name of the program is RELY 4, program no. 620, and there are four different versions of the program. Version 1 and 3 use importance sampling and version 2 and 4 direct simulation . In addition the program provides possibilities of applying different distribution functions for the times to failure and the times to repair.

The computer program RELY 4 calculates both reliability and availability for any system including systems with standby-units and in addition, the standard deviations on both results. The program has been tested on a variety of systems and in all cases the results have agreed with predictions.

2. DESCRIPTION

2.1 General

The program is written in B6700 ALGOL and as already mentioned in the introduction four versions of the program have been worked out, version 1 through 4.

The program is of the Monte Carlo type and its basic principle of operation is very simple:

The times to failure and to repair for each component are simulated - as for version 1 and 3 at increased failure rates - and it is analysed whether the system has failed at any time during the period considered or at the end of the period. This proces is then repeated a certain number of times (F) depending upon the required accuracy of the results. Then it is counted how many times the system has failed 1) at least once during the period considered (FP), 2) at the end of the period (FE). From these figures, as for version 1 and 3 multiplied by weighting factors - the availability of the system at the end of the period considered and the reliability are found as $\frac{FE}{F}$ and $\frac{FP}{F}$ respectively.

The program is based on the assumption that the condition of failure for the syatem to be analysed can be expressed by a boolean expression in terms of failures of a number of units which fail independently with known failure rates.

However, a proper definition of units and formulation of the failure conditions will also make it possible to analyse systems with socalled common mode failures, which is simultaneous failure of two or more units due to a common cause.

Three different categories of units can be considered:

1. Units which fail independently with time dependent probability like a pump for instance.

2. Units which fail independently with time independent probability like an operator for instance.

3. Units that function as standby's with either 50% ("two-out-of-three") or 100% capacity per unit. In the latter case one or two standby's per unit can be considered.

2.2 Flow charts

Fig. 1 and 2 show the flow charts for the two versions (3 and 4) of the program, which are the most generally applicable. In version 3 both the times to failure and the times to repair are exponentially distributed (see section A2, item 1). In version 4 the times to failure and repair are distributed according to a Weibull distribution function (see section A2, item 2). The flow charts for the two other versions of the program (1 and 2) are identical to those for version 3 and 4 respectively with following exceptions:

Version 1: The times to repair for each unit can be distributed according to either a gamma or exponential distribution function (see section A2, item 3). The weighting factors are calculated as described in section A 4.4, item 4).

Version 2: The times to failure and the times to repair are exponentially distributed (see section A2, item 1).

The variables used in the flow charts are described in fig. 4.

The numbers on the flow charts on the right hand side of each block refer to the corresponding point in the program listing in section 2.3.

2.3 Program listing

The following pages contain complete listings of all of the four versions of the program.

Reference points to the corresponding locations in the block diagrams for version 3 and 4 are shown in circles in the right hand side of the listings of version 3 and 4 respectively.

The constants used in the block diagrams are defined in the table on fig. 4.

P 6 2 0 / V 1

```
BEGIN COMMENT ADMA-PROGRAM NR 620/V1-RELY4;

DEFINE                                                              1
PROGRAMNO=620                                                   #,
HEADTEXT=<"PROGRAM NO 620/V1                                   ">#,
FIL5=INP620                                                     #,
FIL6=OUTP620                                                    #;
FILE OUTP620(KIND=70),INP620;


$ INCLUDE "SA/141/START"
 $POP LIST                                                          2
$INCLUDE "SA/107V1"
 $POP LIST                                                          3
BEGIN
INTEGER N,M,F,X,TAELLER,Q,C,P,QHIN,H,L,Y,R,  XORIG,S,G,HSB,  SBK,TUR,NR,   3

K11,E2,Q2,Q12,A10,Q13,Q14,TD,
KL,XNU,AF,Q1,Q2,Q3,Q4,QQPR,QD,LL,Q6,Q7,ND,Q8,Q9,Q10,MP,Q11,NRMK,MAXRES;
REAL GLTID;
READ(INP620,/,N,M,S,        SBK,ND);
BEGIN
INTEGER ARRAY A[0:N+M],NRSB[0:S],HNR[0:S];                          4

BEGIN
LABEL P3,P4,P5,P6,P7,P8,P10,P11,P12;                                5

REAL T,V,U,R,G,VORIG,MAXTID,VAEGTETB,E,D,VB1,T10,T11,YG1,YG2,T12,T13,
SIGMA1R,SIGMA1G,SIGMA2R,SIGMA2G,SIGMA3R,SIGMA3G,HORIG,T9,P221,P211,
L1,L2,Y1,Y2,T1,PR1,PR2,ST1,ST2,P21,P22,VF4,L3,Y3,T2,T3,EPS,HO;
REAL ARRAY FH,YX[0:N],FS[0:M],HM1[0:2*N],HM2[0:2*N],VF2[0:4],VF3[0:4],
LNY[0:N],FNY[0:M],P1,S1,DP1[0:12],VFM[0:N-SBK,0:2*ND],YG[0:N],
VFM1[N-S+1:N,0:4*ND],P2,S2,DP2[0:4];
REAL PROCEDURE RANDOM(X);
INTEGER X;
BEGIN
C:=X*125;                                                           6
X:=C MOD 2796203;
RANDOM:=X/2796203 END RANDOM;
PROCEDURE F1(T2,P1,DP1);                                            6
REAL T2;
REAL ARRAY P1,DP1[0];
BEGIN LL:=LL+1;
DP1[1]:=YG1*P1[8]+Y1*P1[2]+YG2*P1[12]+Y2*P1[3]-L1*P1[1];            6
DP1[2]:=YG2*P1[6]+Y2*P1[4]+L1*P1[1]-(L2+Y1+YG1)*P1[2];
DP1[3]:=YG1*P1[10]+Y1*P1[4]-(Y2+YG2)*P1[3];
DP1[4]:=L2*P1[2]-(Y1+YG1+Y2+YG2)*P1[4];
DP1[5]:=YG2*(P1[4]-P1[5]);
DP1[6]:=YG2*(P1[5]-P1[6]);
DP1[7]:=YG1*(P1[2]-P1[7]);
DP1[8]:=YG1*(P1[7]-P1[8]);
DP1[9]:=YG1*(P1[4]-P1[9]);
DP1[10]:=YG1*(P1[9]-P1[10]);
DP1[11]:=YG2*(P1[3]-P1[11]);
DP1[12]:=YG2*(P1[11]-P1[12]);
END OF F1;
PROCEDURE F2(T2,P2,DP2);                                            6
REAL T2;
REAL ARRAY P2,DP2[0];
BEGIN
```



```
DP2[1]:=Y1*P2[2]+Y2*P2[4]-L1*P2[1];                                 6
DP2[2]:=L1*P2[1]-(Y1+Y2)*P2[2];
DP2[3]:=Y2*(P2[2]-P2[3]);
DP2[4]:=Y2*(P2[3]-P2[4]);
END OF F2;
REAL PROCEDURE VFK1(Q12,A10,T12);                                   6
INTEGER Q12,A10;
REAL T12;
BEGIN
LABEL LA11,LA12;                                                    6

P2[1]:=1;
P2[2]:=0;
P2[3]:=0;
P2[4]:=0;
L1:=FH[Q12];
Y1:=YX[Q12];
Y2:=YG[Q12];
FOR Q:=1 STEP 1 UNTIL 4 DO
S2[Q]:=0;
T2:=0;
HO:=HORIG;
FOR T2:=T2 WHILE T2<T12 DO
BEGIN
IF T2+HO>T12 THEN HO:=T12-T2;                                       7
DIFFSYS(4,T2,P2,HO,EPS,S2,F2,LA11);
LA11:
END;
P21:=P2[1];                                                         7
P22:=P2[2];
P2[1]:=1;
P2[2]:=0;
P2[3]:=0;
P2[4]:=0;
L1:=LNY[Q12];
Y1:=YX[Q12];
Y2:=YG[Q12];
FOR Q:=1 STEP 1 UNTIL 4 DO
S2[Q]:=0;
T2:=0;
HO:=HORIG;
FOR T2:=T2 WHILE T2<T12 DO
BEGIN
IF T2+HO>T12 THEN HO:=T12-T2;                                       7
DIFFSYS(4,T2,P2,HO,EPS,S2,F2,LA12);
LA12:
END;
P211:=P2[1];                                                        7
P221:=P2[2];
IF A10=0 THEN
VFK1:=P21/P211;
IF A10=1 THEN
VFK1:=P22/P221;
END OF VFK1;

REAL PROCEDURE VFS(Q13,Q14);                                        6
INTEGER Q13,Q14;
BEGIN
IF A[Q13]=0 THEN                                                    6
VFS:=VFM[Q13,Q14];
IF A[Q13]=1 THEN
VFS:=VFM[Q13,Q14+ND];
END OF VFS;
REAL PROCEDURE VF1(Q4,MP,T3);                                       6
VALUE Q4;
INTEGER Q4,MP;
REAL T3;
```

```
P1[1]:=1;
P1[2]:=0;
P1[3]:=0;
P1[4]:=0;
P1[5]:=0;
P1[6]:=0;
P1[7]:=0;
P1[8]:=0;
P1[9]:=0;
P1[10]:=0;
P1[11]:=0;
P1[12]:=0;
L1:=FH[NRSB[Q4-N+S]];
L2:=FH[Q4];
LL:=0;
Y1:=YX[NRSB[Q4-N+S]];
Y2:=YX[Q4];
YG1:=YG[NRSB[Q4-N+S]];
YG2:=YG[Q4];
FOR Q:=1 STEP 1 UNTIL 12 DO
S1[Q]:=0;
T2:=0;
HO:=HORIG;
FOR T2:=T2 WHILE T2<T3 DO
BEGIN
IF T2+HO>T3 THEN HO:=T3-T2;                                        7
DIFFSYS(12,T2,P1,HO,EPS,S1,F1,LA7);
LA7:
END;                                                               7
FOR Q:=1 STEP 1 UNTIL 4 DO
VF2[Q]:=P1[Q];
P1[1]:=1;
P1[2]:=0;
P1[3]:=0;
P1[4]:=0;
P1[5]:=0;
P1[6]:=0;
P1[7]:=0;
P1[8]:=0;
P1[9]:=0;
P1[10]:=0;
P1[11]:=0;
P1[12]:=0;
L1:=LNY[NRSB[Q4-N+S]];
L2:=LNY[Q4];
Y1:=YX[NRSB[Q4-N+S]];
Y2:=YX[Q4];
YG1:=YG[NRSB[Q4-N+S]];
YG2:=YG[Q4];
LL:=0;
FOR Q:=1 STEP 1 UNTIL 12 DO
S1[Q]:=0;
T2:=0;
HO:=HORIG;
FOR T2:=T2 WHILE T2<T3 DO
BEGIN
IF T2+HO>T3 THEN HO:=T3-T2;                                        7
DIFFSYS(12,T2,P1,HO,EPS,S1,F1,LA8);
LA8:
END;
FOR Q:=1 STEP 1 UNTIL 4 DO                                         7
VF3[Q]:=P1[Q];
VF1:=VF2[MP]/VF3[MP];
END OF VF1;

REAL PROCEDURE VSB(Q11,K11);                                       6
```



```
INTEGER Q11,K11;
BEGIN
IF A[NRSB[Q11-N+S]]=0 AND A[Q11]=0 THEN ID:=0;                    6
IF A[NRSB[Q11-N+S]]=1 AND A[Q11]=0 THEN ID:=1;
IF A[NRSB[Q11-N+S]]=0 AND A[Q11]=1 THEN ID:=2;
IF A[NRSB[Q11-N+S]]=1 AND A[Q11]=1 THEN ID:=3;
VSB:=VFM1[Q11,K11+ID*ND];
END OF VSB;
REAL PROCEDURE RT(Q10);                                            6
INTEGER Q10;
BEGIN
IF YG[Q10]=0 THEN                                                  6
RT      :=-LN(RANDOM(X))/YX[Q10];
IF YX[Q10]=0 THEN
RT      :=-LN(RANDOM(X))/YG[Q10]-LN(RANDOM(X))/YG[Q10];
END OF RT;
GLTID:=TIME(2);                                                    6
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL N DO FH[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL N DO YX[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL N DO YG[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL S DO NRSB[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL M DO FS[Q]);
READ(INP620,/, FOR Q:=1 STEP 1 UNTIL N DO LNY[Q]);
READ(INP620,/, FOR Q:=1 STEP 1 UNTIL M DO FNY[Q]);
READ(INP620,/,            HO,EPS,T,F,MAXTID,X);
XORIG:=X;
HORIG:=HO;
FOR L:=1 STEP 1 UNTIL N-SBK DO
FOR Q7:=1 STEP 1 UNTIL 2*ND DO
BEGIN
IF Q7 LEQ ND                                                       6
THEN BEGIN T9:=Q7*T/ND;
VFM[L,Q7]:=VFK1(L,0,T9);                                           7
END;
IF Q7>ND THEN                                                      7
BEGIN
T9:=(Q7-ND)*T/ND;                                                  7
VFM[L,Q7]:=VFK1(L,1,T9);
END;
END;                                                               7
FOR L:=N-S+1 STEP 1 UNTIL N DO                                     6
FOR Q7:=1 STEP 1 UNTIL 4*ND DO
BEGIN
IF Q7 LEQ ND THEN                                                  6
BEGIN
T9:=Q7*T/ND;                                                       7
VFM1[L,Q7]:=VF1(L,1,T9);
END;
IF Q7> ND AND Q7 LEQ 2*ND THEN                                     7
BEGIN
T9:=(Q7-ND)*T/ND;                                                  7
VFM1[L,Q7]:=VF1(L,2,T9);
END;
IF Q7>2*ND AND Q7 LEQ 3*ND THEN                                    7
BEGIN
T9:=(Q7-2*ND)*T/ND;                                                7
VFM1[L,Q7]:=VF1(L,3,T9);
END;
IF Q7>3*ND THEN                                                    7
BEGIN
T9:=(Q7-3*ND)*T/ND;                                                7
VFM1[L,Q7]:=VF1(L,4,T9);
END;
END;                                                               7
WRITE(OUTP620,<"TID=",F10.2>,(TIME(2)-GLTID)/60);                  6
E:=0;
D:=0;
F2:=0;
```

```
TAELLER:=0;
SIGMA1R:=0;
SIGMA1G:=0;
SIGMA2R:=0;
SIGMA2G:=0;
SIGMA3R:=0;
SIGMA3G:=0;
P3:
AF:=0;
FOR Q:=N+1 STEP 1 UNTIL N+M DO A[Q]:=0;
FOR Q:=1 STEP 1 UNTIL M DO
IF RANDOM(X) GEQ (1-FNY[Q]) THEN
A[Q+N]:=1;
FOR Q:=1 STEP 1 UNTIL 2*N DO HM2[Q]:=2*T;
V:=0;
FOR Q:=1 STEP 1 UNTIL N-S DO
HM1[Q]:=-LN(RANDOM(X))/LNY[Q];
FOR Q:=1 STEP 1 UNTIL N-S DO
HM1[Q+N]:=HM1[Q]+RT(Q);
FOR Q:=1 STEP 1 UNTIL S DO
BEGIN
HM1[N-S+Q]:=HM1[NRSB[Q]]-LN(RANDOM(X))/LNY[Q];          6
HM1[2*N-S+Q]:=HM1[N-S+Q]+RT(N-S+Q);
END;
P8:  V:=HM1[N+1]; B:=0;                                 6
QMIN:=N+1;
FOR Q:=N+1 STEP 1 UNTIL 2*N DO
IF HM1[Q]<V THEN
BEGIN
V:=HM1[Q];                                              6
QMIN:=Q;
END;
KL:=0;                                                  6
IF V>T THEN
BEGIN
V:=T;                                                   6
KL:=1;
END;
G:=0;                                                   6
TUR:=1;
IF QMIN LEQ(2*N-SBK) THEN
BEGIN
NR:=0;                                                  6
GO TO P5;
END;
FOR H:=1 STEP 1 UNTIL S DO                              6
IF QMIN=NRSB[H]+N
THEN BEGIN
HSB:=H;                                                 6
G:=1;
END;
NR:=0;                                                  6
IF G=1 AND
HM1[N-S+HSB]> HM1[QMIN]
THEN NR:=1;
IF G=1 AND
HM1[N-S+HSB]>HM1[QMIN-N]
AND
HM1[N-S+HSB]<HM1[QMIN]
THEN NR:=2;
IF G=1 AND
HM1[N-S+HSB]<HM1[QMIN-N]
THEN NR:=3;
IF G=0 AND
HM1[NRSB[QMIN-2*N+S]]<HM1[QMIN]
AND
HM1[NRSB[QMIN-2*N+S]]>HM1[QMIN-N]
```



```
THEN NR:=4;
IF G=0 AND
HM1[QMIN-N]>HM1[NRSB[QMIN-2*N+S]]
THEN NR:=5;
IF G=0
AND
HM1[NRSB[QMIN-2*N+S]]>HM1[QMIN]
THEN NR:=6;
IF NR=1 OR NR=2 THEN
BEGIN VORIG:=V;
V:=HM1[QMIN-N]+1.0P-8;                                  6
END;
IF NR=5 THEN                                            6
BEGIN
VORIG:=V;                                               6
V:=HM1[NRSB[QMIN-2*N+S]]+1.0P-8;
END;
IF NR=6 THEN                                            6
BEGIN VORIG:=V;
V:=HM1[QMIN-N]+1.0P-8;                                  6
END;
IF V > T+1.0P-9 THEN GO TO P7;                          6
P5:
FOR P:=1 STEP 1 UNTIL N DO
A[P]:=0;
FOR H:=1 STEP 1 UNTIL N DO
IF HM1[H]<V
OR(HM2[H]<V AND HM2[H+N]>V)THEN
A[H]:=1;
B:=0;
IF

{ A[1]=1 AND A[2]=1 }   INPUT: System failure conditions

THEN BEGIN
B:=1;                                                   6
AF:=AF+1;
END;
IF B=1 THEN Y:=1;                                       6
IF B=1 AND(AF=1 OR KL=1) THEN
BEGIN
VAEGTETB:=1;                                            6
QB:=ENTIER(V*ND/T+0.5);
IF QB=0 THEN QB:=1;
FOR Q:=1 STEP 1 UNTIL N-SBK DO
VAEGTETB:=VAEGTETB*VFS(Q,QB);
FOR P:=N-S+1 STEP 1 UNTIL N DO
BEGIN
Q3:=0;                                                  7
FOR Q2:=1 STEP 1 UNTIL S DO
IF P= NRSB[Q2] THEN Q3:=1;
IF Q3=0 THEN
VAEGTETB:=VAEGTETB*VSB(P,QB);
END;
FOR Q:=1 STEP 1 UNTIL N DO                              7
BEGIN
IF A[Q+N]=1 THEN                                        7
VAEGTETB:=VAEGTETB*FS[Q]/FNY[Q];
IF A[Q+N]=0 THEN
VAEGTETB:=VAEGTETB*(1-FS[Q])/(1-FNY[Q]);
END;
END;                                                    7
P7:                                                     6
IF AF=1 THEN VB1:=VAEGTETB;
IF(NR=1 OR NR=2 OR NR=5 OR NR=6) AND TUR=1
THEN
BEGIN
V:=VORIG;
```

```
GO TO P5;
END;
```



```
IF KL=1 THEN GO TO P6;                                                6
FOR Q:=1 STEP 1 UNTIL 2*N DO
HM2[Q]:=HM1[Q];
IF NR=0 OR NR=3 OR NR=4 OR NR=5 THEN
BEGIN
HM1[QMIN-N]:=HM1[QMIN]-LN(RANDOM(X))/LNY[QMIN-N];                     6
HM1[QMIN]:=HM1[QMIN-N]+RT(QMIN-N);
END;
IF NR=1 THEN                                                          6
BEGIN
HM1[QMIN-N]:=HM1[N-S+HSB]-LN(RANDOM(X))/LNY[QMIN-N];                  6
HM1[QMIN]:=HM1[QMIN-N]+RT(QMIN-N);
END;
IF NR=2 THEN                                                          6
BEGIN
HM1[QMIN-N]:=HM1[QMIN]-LN(RANDOM(Y))/LNY[QMIN-N];                     6
HM1[QMIN]:=HM1[QMIN-N]+RT(QMIN-N);
FOR Q:=N+1 STEP 1 UNTIL N+M DO A[Q]:=0;
FOR Q:=1 STEP 1 UNTIL M DO
IF RANDOM(X) GEQ (1-FNY[Q]) THEN
A[Q+N]:=1;
END;
IF NR=6 THEN                                                          6
BEGIN
HM1[QMIN-N]:=HM1[NRSB[QMIN-2*N+S]]-LN(RANDOM(X))/LNY[QMIN-N];         6
HM1[QMIN]:=HM1[QMIN-N]+RT(QMIN-N);
FOR Q:=N+1 STEP 1 UNTIL N+M DO A[Q]:=0;
FOR Q:=1 STEP 1 UNTIL M DO
IF RANDOM(X) GEQ (1-FNY[Q]) THEN
A[Q+M]:=1;
END;
P6:                                                                   6
IF KL=0 THEN
GO TO  P8;
IF B=1
THEN D:=D+VAEGTETR;
IF Y=1 THEN E:=E+VB1;
IF Y=1 THEN
E2:=E2+1;
IF B=1 THEN
D2:=D2+1;
TAELLER:=TAELLER+1;
SIGMA1R:=SIGMA1R+(VB1*Y)**2;
SIGMA1G:=SIGMA1G+(VAEGTETR*B)**2;
IF TAELLER<F AND
MAXTID*60-TIME(2)/60 > 15 THEN
GO TO P3;
R:=(TAELLER-E)/TAELLER;
G:=(TAELLER-D)/TAELLER;
SIGMA2R:=(SQRT(SIGMA1R/TAELLER-(1-R)**2))/SQRT(TAELLER);
SIGMA2G:=(SQRT(SIGMA1G/TAELLER-(1-G)**2))/SQRT(TAELLER);
WRITE(OUTP620,<"DIVERSE INPUT STOERRELSER">);
WRITE(OUTP620,<"N=",J8>,N);
WRITE(OUTP620,<"M=",J8>,M);
WRITE(OUTP620,<"S=",J8>,S);
WRITE(OUTP620,<"ANTAL STANDBYKOBLEDE KOMPONENTER=",J8>,SBK);
WRITE(OUTP620,<"ANTAL TIDSSTEP FOR VAEGTFAKTOR=",J8>,ND);
WRITE(OUTP620,<"FEJLHYPPIGHEDER I TIMER**(-1)">);
FOR Q:=1 STEP 1 UNTIL N DO
WRITE(OUTP620,<"FH[",J5,"]=",E11.4>,Q,FH[Q]);
WRITE(OUTP620,<"REPARATIONSHYPPIGHEDER I TIMER**(-1)">);
WRITE(OUTP620,<"A)  MED EKSPONENTIEL FORDELING:">);
FOR Q:=1 STEP 1 UNTIL N DO
WRITE(OUTP620,<"YX[",J5,"]=",E11.4>,Q,YX[Q]);
WRITE(OUTP620,<"B)  MED GAMMA FORDELING:">);
```

```
FOR Q:=1 STEP 1 UNTIL N DO
```



```
WRITE(OUTP620,<"YG[",J5,"]=",E11.4>,Q,YG[Q]);
WRITE(OUTP620,<"DRIFTSKOMPONENT",X5,"STANDBYKOMPONENT">);
FOR Q:=1 STEP 1 UNTIL S DO
WRITE(OUTP620,<X4,I4,X17,I4>,NRSB[Q],N-S+Q);
WRITE(OUTP620,<"FEJLSANDSYNLIGHEDER">);
FOR Q:=1 STEP 1 UNTIL M DO
WRITE(OUTP620,<"FS[",J5,"]=",E11.4>,Q+N,FS[Q]);
WRITE(OUTP620,<"KORRIGEREDE FEJLHYPPIGHEDER:">);
FOR Q:=1 STEP 1 UNTIL N DO
WRITE(OUTP620,<"LNY[",J5"]=",E11.4>,Q,LNY[Q]);
WRITE(OUTP620,<"KORRIGEREDE FEJLSANDSYNLIGHEDER:">);
FOR Q:=1 STEP 1 UNTIL M DO
WRITE(OUTP620,<"FNY[",J5"]=",E11.4>,Q+N,FNY[Q]);
WRITE(OUTP620,<"INPUT TIL DIFFSYS:">);
WRITE(OUTP620,<"HO=",E11.4>,HORIG);
WRITE(OUTP620,<"EPS=",E11.4>,EPS);
WRITE(OUTP620,<"LL=",J8>,LL);
WRITE(OUTP620,<"T=",F12.2,"TIMER">,T);
WRITE(OUTP620,<"F=",J10>,F);
WRITE(OUTP620,<"MAX KOERETID I MINUTTER=",J8>,MAXTID);
WRITE(OUTP620,<"X=",J8>,XORIG);
WRITE(OUTP620,<"RESULTATER:">);
WRITE(OUTP620,<"ANTAL KOERTE MONTE CARLO FORSOEG=",J10>,TAELLER);
WRITE(OUTP620,<"FINGERET PAALIDELIGHED=",E11.4>,(TAELLER-E2)/TAELLER);
WRITE(OUTP620,<"FINGERET AVAILABILITY TIL TIDEN T=",E11.4>,
(TAELLER-D2)/TAELLER);
WRITE(OUTP620,<"PAALIDELIGHEDEN AF SYSTEMET ER LIG MED",F15.12>,R);
WRITE(OUTP620,<"STANDARDAFVIGELSE PAA PAALIDELIGHEDEN=",F15.12>,
SIGMA2R);
WRITE(OUTP620,<"AVAILABILITY TIL TIDSPUNKTET T ER LIG MED",F15.12>,G);
WRITE(OUTP620,<"STANDARDAFVIGELSE PAA AVAILABILITY=",X6,F15.12>,
SIGMA2G);
WRITE(OUTP620,<"MED 95 PROCENT KONFIDENS ER SYSTEMETS FEJLSANDSYNLIGHED
">);

WRITE (OUTP620,<"LIG MED ",   E11.4,"PLUS MINUS",  E11.4>,1-R,2*
SIGMA2R);
END OF P620;



END;

END;

$ INCLUDE "SA/141/SLUT"
 $PUP LIST
END,
```

P 6 2 0 / V 2
= = = = = = =

```
BEGIN COMMENT ADM4-PROGRAM NR 620/V2-RELY4;

DEFINE                                                                   1
PROGRAMNO=620                                                    #,
HEADTEXT=<"PROGRAM NO 620/V2                                    ">#,
FIL5=INP620                                                      #,
FIL6=OUTP620                                                     #;
FILE OUTP620(KIND=70),INP620;


$ INCLUDE "SA/141/START"
 $POP LIST                                                               2
BEGIN
INTEGER N,M,F,X,YAELLER,Q,C,P,QMIN,H,L,Y,B,E,XORIG,S,G,HSB,D,SBK,TUR,NR,   3

KL,XNU;
REAL GLTID;
READ(INP620,/,N,M,S,SBK);
BEGIN
INTEGER ARRAY A[0:N+M],NRSB[0:S];                                        4

BEGIN
LABEL P1,P2,P3,P4,P5,P6,P7;                                              5

REAL T,V,U,R,G1,VORIG,MAXTID;
REAL ARRAY FH,RH[0:N],FS[0:M],HM1[0:2*N],HM2[0:2*N];
REAL PROCEDURE RANDOM(X);
INTEGER X;
BEGIN
C:=X*125;                                                                6
X:=C MOD 2796203;
RANDOM:=X/2796203 END RANDOM;
GLTID:=TIME(2);                                                          6
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL N DO FH[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL N DO RH[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL S DO NRSB[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL M DO FS[Q]);
READ(INP620,/,T,F,MAXTID,X);
XORIG:=X;
E:=0;
D:=0;
TAELLER:=0;
P3:
FOR Q:=N+1 STEP 1 UNTIL N+M DO A[Q]:=0;
FOR Q:=1 STEP 1 UNTIL M DO
IF RANDOM(X) GEQ (1-FS[Q]) THEN
A[Q+N]:=1;
FOR Q:=1 STEP 1 UNTIL 2*N DO HM2[Q]:=2*T;
Y:=0;
FOR Q:=1 STEP 1 UNTIL N-S DO
HM1[Q]:=-LN(RANDOM(X))/FH[Q];
FOR Q:=1 STEP 1 UNTIL N-S DO
HM1[Q+N]:=HM1[Q]-LN(RANDOM(X))/RH[Q];
FOR Q:=1 STEP 1 UNTIL S DO
BEGIN
HM1[N-S+Q]:=HM1[NRSB[Q]]-LN(RANDOM(X))/FH[N-S+Q];                        6
HM1[2*N-S+Q]:=HM1[N-S+Q]-LN(RANDOM(X))/RH[N-S+Q];
END;
P1:  V:=HM1[N+1]; B:=0;                                                  6
QMIN:=N+1;
```



```
FOR Q:=N+1 STEP 1 UNTIL 2*N DO
IF HM1[Q]<V THEN
BEGIN
V:=HM1[Q];                                                               6
QMIN:=Q;
END;
KL:=0;                                                                   6
IF V>T THEN
BEGIN
V:=T;                                                                    6
KL:=1;
END;
G:=0;                                                                    6
TUR:=1;
IF QMIN LEQ(2*N-SBK) THEN
BEGIN
NR:=0;                                                                   6
GO TO P5;
END;
FOR H:=1 STEP 1 UNTIL S DO                                               6
IF QMIN=NRSB[H]+N
THEN BEGIN
HSB:=H;                                                                  6
G:=1;
END;
NR:=0;                                                                   6
IF G=1 AND
HM1[N-S+HSB]> HM1[QMIN]
THEN NR:=1;
IF G=1 AND
HM1[N-S+HSB]>HM1[QMIN-N]
AND
HM1[N-S+HSB]<HM1[QMIN]
THEN NR:=2;
IF G=1 AND
HM1[N-S+HSB]<HM1[QMIN-N]
THEN NR:=3;
IF G=0 AND
HM1[NRSB[QMIN-2*N+S]]<HM1[QMIN]
AND
HM1[NRSB[QMIN-2*N+S]]>HM1[QMIN-N]
THEN NR:=4;
IF G=0 AND
HM1[QMIN-N]>HM1[NRSB[QMIN-2*N+S]]
THEN NR:=5;
IF G=0
AND
HM1[NRSB[QMIN-2*N+S]]>HM1[QMIN]
THEN NR:=6;
IF NR=1 OR NR=2 THEN
BEGIN VORIG:=V;
V:=HM1[QMIN-N]+1,0@-8;                                                   6
END;
IF NR=5 THEN                                                             6
BEGIN
VORIG:=V;                                                                6
V:=HM1[NRSB[QMIN-2*N+S]]+1,0@-8;
END;
IF NR=6 THEN                                                             6
BEGIN VORIG:=V;
V:=HM1[QMIN-N]+1,0@-8;                                                   6
END;
IF V > T+1,0@-9 THEN GO TO P7;                                           6
P5:
FOR P:=1 STEP 1 UNTIL N DO
A[P]:=0;
FOR H:=1 STEP 1 UNTIL N DO
IF HM1[H]<V
```

```
A[M]:=1;
Q:=0;
IF
A[1]=1 AND A[2]=1      INPUT: System failure conditions
THEN B:=1;
IF B=1 THEN Y:=1;
P7:
IF(NR=1 OR NR=2 OR NR=5 OR NR=6) AND TUR=1
THEN
BEGIN
V:=VORIG;                                                              6
TUR:=2;
GO TO P5;
END;
IF KL=1 THEN GO TO P6;                                                 6
FOR Q:=1 STEP 1 UNTIL 2*N DO
HM2[Q]:=HM1[Q];
IF NR=0 OR NR=3 OR NR=4 OR NR=5 THEN
BEGIN
HM1[QMIN-N]:=HM1[QMIN]-LN(RANDOM(X))/FH[QMIN-N];                       A
HM1[QMIN]:=HM1[QMIN-N]-LN(RANDOM(X))/RH[QMIN-N];
END;
IF NR=1 THEN                                                           6
BEGIN
HM1[QMIN-N]:=HM1[N-S+HSB]-LN(RANDOM(X))/FH[QMIN-N];                    6
HM1[QMIN]:=HM1[QMIN-N]-LN(RANDOM(X))/RH[QMIN-N];
END;
IF NR=2 THEN                                                           6
BEGIN
HM1[QMIN-N]:=HM1[QMIN]-LN(RANDOM(X))/FH[QMIN-N];                       6
HM1[QMIN]:=HM1[QMIN-N]-LN(RANDOM(X))/RH[QMIN-N];
FOR Q:=N+1 STEP 1 UNTIL N+M DO A[Q]:=0;
FOR Q:=1 STEP 1 UNTIL M DO
IF RANDOM(X) GEQ (1-FS[Q]) THEN
A[Q+N]:=1;
END;
IF NR=6 THEN                                                           6
BEGIN
HM1[QMIN-N]:=HM1[NRSB[QMIN-2*N+S]]-LN(RANDOM(X))/FH[QMIN-N];           6
HM1[QMIN]:=HM1[QMIN-N]-LN(RANDOM(X))/RH[QMIN-N];
FOR Q:=N+1 STEP 1 UNTIL N+M DO A[Q]:=0;
FOR Q:=1 STEP 1 UNTIL M DO
IF RANDOM(X) GEQ (1-FS[Q]) THEN
A[Q+N]:=1;
END;
P6:                                                                    6
IF KL=0 THEN
GO TO P1;
IF B=1
THEN D:=D+1;
IF Y=1 THEN E:=E+1;
TAELLER:=TAELLER+1;
IF TAELLER<F AND
MAXTID*60-TIME(2)/60 > 15 THEN
GO TO P3;
R:=(TAELLER-E)/TAELLER;
G1:=(TAELLER-D)/TAELLER;
WRITE(OUTP620,<"DIVERSE INPUT STOERRELSER">);
WRITE(OUTP620,<"N=",J8>,N);
WRITE(OUTP620,<"M=",J8>,M);
WRITE(OUTP620,<"S=",J8>,S);
WRITE(OUTP620,<"ANTAL STANDBYKOBLEDE KOMPONENTER=",J8>,SBK);
WRITE(OUTP620,<"FEJLHYPPIGHEDER I TIMER**(-1)">);
FOR Q:=1 STEP 1 UNTIL N DO
WRITE(OUTP620,<"FH[",J5,"]=",E11.4>,Q,FH[Q]);
WRITE(OUTP620,<"REPARATIONSHYPPIGHEDER I TIMER**(-1)">);
FOR Q:=1 STEP 1 UNTIL N DO
```



```
WRITE(OUTP620,<"RH[",J5,"]=",E11.4>,Q,RH[Q]);
WRITE(OUTP620,<"DRIFTSKOMPONENT",X5,"STANDBYKOMPONENT">);
FOR Q:=1 STEP 1 UNTIL S DO
WRITE(OUTP620,<X4,I4,X17,I4>,NRSB[Q],N-S+Q);
WRITE(OUTP620,<"FEJLSANDSYNLIGHEDER">);
FOR Q:=1 STEP 1 UNTIL M DO
WRITE(OUTP620,<"FS[",J5,"]=",E11.4>,Q+N,FS[Q]);
WRITE(OUTP620,<"T=",F12.2,"TIMER">,T);
WRITE(OUTP620,<"F=",J10>,F);
WRITE(OUTP620,<"MAX KOERETID I MINUTTER=",J8>,MAXTID);
WRITE(OUTP620,<"X=",J8>,XORIG);
WRITE(OUTP620,<"RESULTATER:">);
WRITE(OUTP620,<"ANTAL KOERTE MONTE CARLO FORSOEG=",J10>,TAELLER);
WRITE(OUTP620,<"PAALIDELIGHEDEN AF SYSTEMET ER LIG MED",F15.12>,R);
WRITE(OUTP620,<"STANDARDAFVIGELSE PAA PAALIDELIGHEDEN=",F15.12>,
SQRT(R*(1-R)/TAELLER));
WRITE(OUTP620,<"AVAILABILITY TIL TIDSPUNKTET T ER LIG MED",F15.12>,G1);
WRITE(OUTP620,<"STANDARDAFVIGELSE PAA AVAILABILITY=",X6,F15.12>,
SQRT(G1*(1-G1)/TAELLER));
END OF P620;

WRITE(OUTP620,<"TID=",F10.2>,(TIME(2)-GLTID)/60);                      5
END;

END;                                                                   4

£ INCLUDE "SA/141/SLUT"                                                3
 $PUP LIST                                                             2
END.
```

P 6 2 0 / V 3
= = = = = = =

(1)

```
BEGIN COMMENT ADMA-PROGRAM NR 620/V2-RELYA;

DEFINE                                                              1
PROGRAMNO=620                                                  #,
HEADTEXT=<"PROGRAM NO 620/V3                                  ">#,
FIL5=INP620                                                    #,
FIL6=OUTP620                                                   #;
FILE OUTP620(KIND=70),INP620;


$ INCLUDE "SA/141/START"                                            2
 $POP LIST
$INCLUDE "SA/107V1"                                                 3
 $POP LIST
BEGIN
INTEGER N,M,F,X,TAELLER,Q,C,P,QMIN,H,L,Y,B,  XORIG,S,G,HSB,  SBK,TUR,NR,   3

K11,E2,D2,NUMMER1,NUMMER2,MINNR,PS,Q14,K14,K15,Q15,MPS,Q16,Q17,K16,Q18,
Q19,
KL,XNU,AF,Q1,Q2,Q3,Q4,QDPR,QD,LL,Q6,Q7,ND,Q8,Q9,Q10,MP,Q11,NRMK,MAXRFS;
REAL GLTID,MINT,T4,T5,T6;
READ(INP620,/,N,M,S,PS,    SBK,ND);
BEGIN
INTEGER ARRAY A[0:N+M],NRSB[0:S],HNR[0:S];                         4

BEGIN
LABEL    P4,P5,P6,P7,P8,P10,P11,P12,P13,P14,P15;                   5

REAL T,V,U,R,G1,VORIG,MAXTID,VAEGTETR,E,D,        VB1,T10,T11,
SIGMA1R,SIGMA1G,SIGMA2R,SIGMA2G,SIGMA3R,SIGMA3G,HORIG,T9,
L1,L2,Y1,Y2,T1,PR1,PR2,ST1,ST2,              L3,Y3,T2,T3,EPS,H0;
REAL ARRAY FH,RH[0:N],FS[0:M],HM1[0:2*N],HM2[0:2*N],VF2[0:12],
VF3[0:12],        VFT1[0:S,0:ND],VFT2[0:S,0:ND],VFT3[0:S,0:ND],
VFT4[0:S,0:ND],VFT5[0:S,0:ND],VFT6[0:S,0:ND],VFT7[0:S,0:ND],VFT8[0:S,
0:ND],VFT9[0:S,0:ND],LNY[0:N],FNY[0:M],
                    VFT10[0:S,0:ND],VFT11[0:S,0:ND],VFT12[0:S,0:ND],
P1,S1,DP1[0:4],P2,S2,DP2[0:8],VFT[0:S,0:ND],VFT0[0:S,0:NO],P3 PF,PFY,S3,
DP3[0:8],PFT1[0:PS,0:ND],PFT2[0:PS,0:ND],PFT3[0:PS,0:ND],
PFT4[0:PS,0:ND],PFT5[0:PS,0:ND],PFT6[0:PS,0:ND],PFT7[0:PS,0:ND],
PFT8[0:PS,0:ND];
REAL PROCEDURE RANDOM(X);
INTEGER X;
BEGIN
C:=X*125;                                                           6
X:=C MOD 2796203;
RANDOM:=X/2796203 END RANDOM;
PROCEDURE F2(T2,P2,DP2);                                            6
REAL T2;
REAL ARRAY P2,DP2[0];
BEGIN
LL:=LL+1;                                                           6
DP2[1]:=Y1*P2[2]+Y2*P2[3]+Y3*P2[4]-L1*P2[1];
DP2[2]:=Y2*P2[5]+Y3*P2[6]+L1*P2[1]-(L2+Y1)*P2[2];
DP2[3]:=Y1*P2[5]-Y2*P2[3]+Y3*P2[7];
DP2[4]:=Y2*P2[7]+Y1*P2[6]-Y3*P2[4];
DP2[5]:=Y3*P2[8]+L2*P2[2]-(L3+Y1+Y2)*P2[5];
DP2[6]:=Y2*P2[8]-(Y3+Y1)*P2[6];
DP2[7]:=Y1*P2[8]-(Y2+Y3)*P2[7];
DP2[8]:=L3*P2[5]-(Y1+Y2+Y3)*P2[8];
END OF F2;
```



```
PROCEDURE F1(T2,P1,DP1);                                            6
REAL T2;
REAL ARRAY P1,DP1[0];
BEGIN LL:=LL+1;
DP1[1]:=Y2*P1[3]+Y1*P1[2]-L1*P1[1];                                 6
DP1[2]:=Y2*P1[4]+L1*P1[1]-(Y1+L2)*P1[2];
DP1[3]:=Y1*P1[4]-Y2*P1[3];
DP1[4]:=L2*P1[2]-(Y1+Y2)*P1[4];
END OF F1;
     PROCEDURE VF1(Q4,T3);                                          6
VALUE Q4;
INTEGER Q4;
REAL T3;
BEGIN
LABEL LA7,LA8,LA9,LA10;                                             6

IF H=1 THEN
BEGIN
P1[1]:=1;                                                           7
P1[2]:=0;
P1[3]:=0;
P1[4]:=0;
L1:=FH[NRSB[Q4-N+S]];
L2:=FH[Q4];
LL:=0;
Y1:=RH[NRSB[Q4-N+S]];
Y2:=RH[Q4];
FOR Q:=1 STEP 1 UNTIL 4 DO
S1[Q]:=0;
T2:=0;
H0:=HORIG;
FOR T2:=T2 WHILE T2<T3 DO
BEGIN
IF T2+H0>T3 THEN H0:=T3-T2;                                         8
DIFFSYS(4,T2,P1,H0,EPS,S1,F1,LA7);
LA7;
END;
FOR Q:=1 STEP 1 UNTIL 4                                             8
DO VF2[Q]:=P1[Q];
P1[1]:=1;
P1[2]:=0;
P1[3]:=0;
P1[4]:=0;
L1:=LNY[NRSB[Q4-N+S]];
L2:=LNY[Q4];
Y1:=RH[NRSB[Q4-N+S]];
Y2:=RH[Q4];
LL:=0;
FOR Q:=1 STEP 1 UNTIL 4 DO
S1[Q]:=0;
T2:=0;
H0:=HORIG;
FOR T2:=T2 WHILE T2<T3 DO
BEGIN
IF T2+H0>T3 THEN H0:=T3-T2;                                         8
DIFFSYS(4,T2,P1,H0,EPS,S1,F1,LA8);
LA8;
END;
FOR Q:=1 STEP 1 UNTIL 4                                             8
DO VF3[Q]:=P1[Q];
END;
IF H=2 THEN                                                         7
BEGIN
P2[1]:=1;                                                           7
P2[2]:=0;
P2[3]:=0;
P2[4]:=0;
P2[5]:=0;
```

```
P2[7]:=0;
P2[8]:=0;
L1:=FH[NRSB[NRSB[Q4-N+S]-N+S]];
L2:=FH[NRSB[Q4-N+S]];
L3:=FH[Q4];
Y1:=RH[NRSB[NRSB[Q4-N+S]-N+S]];
Y2:=RH[NRSB[Q4-N+S]];
Y3:=RH[Q4];
LL:=0;
FOR Q:=1 STEP 1 UNTIL 8 DO
S2[Q]:=0;
T2:=0;
HO:=HORIG;
FOR T2:=T2 WHILE T2<T3 DO
BEGIN                                                    8
IF T2+HO>T3 THEN HO:=T3-T2;
DIFFSYS(8,T2,P2,HO,EPS,S2,F2,LA9);
LA9:
END;                                                     8
FOR Q:=1 STEP 1 UNTIL 8 DO
VF2[Q+4]:=P2[Q];
P2[1]:=1;
P2[2]:=0;
P2[3]:=0;
P2[4]:=0;
P2[5]:=0;
P2[6]:=0;
P2[7]:=0;
P2[8]:=0;
L1:=LNY[NRSB[NRSB[Q4-N+S]-N+S]];
L2:=LNY[NRSB[Q4-N+S]];
L3:=LNY[Q4];
Y1:=RH[NRSB[NRSB[Q4-N+S]-N+S]];
Y2:=RH[NRSB[Q4-N+S]];
Y3:=RH[Q4];
LL:=0;
FOR Q:=1 STEP 1 UNTIL 8 DO
S2[Q]:=0;
T2:=0;
HO:=HORIG;
FOR T2:=T2 WHILE T2<T3 DO
BEGIN                                                    8
IF T2+HO>T3 THEN HO:=T3-T2;
DIFFSYS(8,T2,P2,HO,EPS,S2,F2,LA10);
LA10:
END;
FOR Q:=1 STEP 1 UNTIL 8 DO                               8
VF3[Q+4]:=P2[Q];
END;
END OF VF1;                                              7

REAL                                                     6
PROCEDURE VFK(MP,Q10,T10);
INTEGER MP,Q10;
REAL T10;
BEGIN
IF MP<5 THEN                                             6
H:=1 ELSE H:=2;
VF1(Q10,T10);
VFK:=VF2[MP]/VF3[MP];
END OF VFK;
REAL PROCEDURE VSB(Q11,K11);                             6
VALUE Q11;
INTEGER Q11,K11;
BEGIN
LABEL P9;                                                6
```



```
QOPR:=Q11;
H:=1;
P9:
IF NRSB[Q11-N+S] > N-S THEN
BEGIN
H:=H+1;                                                  7
Q11:=NRSB[Q11-N+S];
GO TO P9;
END;
Q11:=QOPR;                                               7
IF H=1 THEN
BEGIN
IF A[Q11]=0 AND                                          7
A[NRSB[Q11-N+S]]=0 THEN
NRMK:=1;
IF A[Q11]=0 AND
A[NRSB[Q11-N+S]]=1 THEN
NRMK:=2;
IF A[Q11]=1 AND
A[NRSB[Q11-N+S]]=0 THEN
NRMK:=3;
IF A[Q11]=1 AND
A[NRSB[Q11-N+S]]=1 THEN
NRMK:=4;
END;
IF H=2 THEN                                              7
BEGIN
IF A[Q11]=0 AND A[NRSB[Q11-N+S]]=0 AND A[NRSB[NRSB[Q11-N+S]-N+S]]=0     7
THEN NRMK:=5 ;
IF A[Q11]=0 AND A[NRSB[Q11-N+S]]=0 AND A[NRSB[NRSB[Q11-N+S]-N+S]]=1
THEN NRMK:=6 ;
IF A[Q11]=0 AND A[NRSB[Q11-N+S]]=1 AND A[NRSB[NRSB[Q11-N+S]-N+S]]=0
THEN NRMK:=7 ;
IF A[Q11]=1 AND A[NRSB[Q11-N+S]]=0 AND A[NRSB[NRSB[Q11-N+S]-N+S]]=0
THEN NRMK:=8 ;
IF A[Q11]=0 AND A[NRSB[Q11-N+S]]=1 AND A[NRSB[NRSB[Q11-N+S]-N+S]]=1
THEN NRMK:=9 ;
IF A[Q11]=1 AND A[NRSB[Q11-N+S]]=0 AND A[NRSB[NRSB[Q11-N+S]-N+S]]=1
THEN NRMK:=10;
IF A[Q11]=1 AND A[NRSB[Q11-N+S]]=1 AND A[NRSB[NRSB[Q11-N+S]-N+S]]=0
THEN NRMK:=11;
IF A[Q11]=1 AND A[NRSB[Q11-N+S]]=1 AND A[NRSB[NRSB[Q11-N+S]-N+S]]=1
THEN NRMK:=12;
END;
IF NRMK=1 THEN                                           7
VSB:=VFT1[Q11-N+S,K11];
IF NRMK=2  THEN
VSB:=VFT2 [Q11-N+S,K11];
IF NRMK=3  THEN
VSB:=VFT3 [Q11-N+S,K11];
IF NRMK=4  THEN
VSB:=VFT4 [Q11-N+S,K11];
IF NRMK=5  THEN
VSB:=VFT5 [Q11-N+S,K11];
IF NRMK=6  THEN
VSB:=VFT6 [Q11-N+S,K11];
IF NRMK=7  THEN
VSB:=VFT7 [Q11-N+S,K11];
IF NRMK=8  THEN
VSB:=VFT8 [Q11-N+S,K11];
IF NRMK=9  THEN
VSB:=VFT9 [Q11-N+S,K11];
IF NRMK=10 THEN
VSB:=VFT10[Q11-N+S,K11];
IF NRMK=11 THEN
VSB:=VFT11[Q11-N+S,K11];
IF NRMK=12 THEN
VSB:=VFT12[Q11-N+S,K11];
```

```
PROCEDURE F3(T4,P3,DP3);                                        6
REAL T4;
REAL ARRAY P3,DP3[0];
BEGIN
DP3[1]:=Y1*P3[2]+Y2*P3[3]+Y3*P3[4]-(L1+L2)*P3[1];               6
DP3[2]:=L1*P3[1]+Y2*P3[5]+Y3*P3[6]-(L2+L3+Y1)*P3[2];
DP3[3]:=L2*P3[1]+Y1*P3[5]+Y3*P3[7]-(Y2+L1+L3)*P3[3];
DP3[4]:=Y1*P3[6]+Y2*P3[7]-(Y3+L1+L2)*P3[4];
DP3[5]:=L2*P3[2]+L1*P3[3]+Y3*P3[8]-(Y2+Y1+L3)*P3[5];
DP3[6]:=L3*P3[2]+L1*P3[4]+Y2*P3[8]-(Y3+Y1+L2)*P3[6];
DP3[7]:=L2*P3[4]+L3*P3[3]+Y1*P3[8]-(Y3+Y2+L1)*P3[7];
DP3[8]:=L3*P3[5]+L2*P3[6]+L1*P3[7]-(Y1+Y2+Y3)*P3[8];
END OF F3;
PROCEDURE VF4(Q15,T5);                                          6
VALUE Q15;
INTEGER Q15;
REAL T5;
BEGIN
LABEL LA11,LA12;                                                6

P3[1]:=1;
P3[2]:=0;
P3[3]:=0;
P3[4]:=0;
P3[5]:=0;
P3[6]:=0;
P3[7]:=0;
P3[8]:=0;
L1:=FH[PS+2*(Q15-1)+1];
L2:=FH[PS+2*(Q15-1)+2];
L3:=FH[Q15];
Y1:=RH[PS+2*(Q15-1)+1];
Y2:=RH[PS+2*(Q15-1)+2];
Y3:=RH[Q15];
FOR Q:=1 STEP 1 UNTIL 8 DO
S3[Q]:=0;
T2:=0;
H0:=HORIG;
FOR T2:=T2 WHILE T2<T5 DO
BEGIN
IF T2+H0>T5 THEN H0:=T5-T2;                                     7
DIFFSYS(8,T2,P3,H0,EPS,S3,F3,LA11);
LA11:
END;
FOR Q:=1 STEP 1 UNTIL 8 DO                                      7
PF[Q]:=P3[Q];
P3[1]:=1;
P3[2]:=0;
P3[3]:=0;
P3[4]:=0;
P3[5]:=0;
P3[6]:=0;
P3[7]:=0;
P3[8]:=0;
L1:=LNY[PS+2*(Q15-1)+1];
L2:=LNY[PS+2*(Q15-1)+2];
L3:=LNY[Q15];
Y1:=RH[PS+2*(Q15-1)+1];
Y2:=RH[PS+2*(Q15-1)+2];
Y3:=RH[Q15];
FOR Q:=1 STEP 1 UNTIL 8 DO
S3[Q]:=0;
T2:=0;
H0:=HORIG;
FOR T2:=T2 WHILE T2<T5 DO
BEGIN
```



```
IF T2+H0>T5 THEN H0:=T5-T2;                                     7
DIFFSYS(8,T2,P3,H0,EPS,S3,F3,LA12);
LA12:
END;
FOR Q:=1 STEP 1 UNTIL 8 DO                                      7
PFY[Q]:=P3[Q];
END OF VF4;

REAL PROCEDURE PFK(NPS,Q16,T6);                                 6
INTEGER NPS,Q16;
REAL T6;
BEGIN
VF4(Q16,T6);                                                    6
PFK:=PF[NPS]/PFY[NPS];
END OF PFK;
REAL PROCEDURE PSB(Q17,K16);                                    6
VALUE Q17;
INTEGER Q17,K16;
BEGIN
LABEL P14;                                                      6

IF A[Q17]=0 AND A[PS+2*(Q17-1)+1]=0 AND A[PS+2*(Q17-1)+2]=0
THEN
PSB:=PFT1[Q17,K16];
IF A[Q17]=0 AND A[PS+2*(Q17-1)+1]=1 AND A[PS+2*(Q17-1)+2]=0
THEN
PSB:=PFT2[Q17,K16];
IF A[Q17]=0 AND A[PS+2*(Q17-1)+1]=0 AND A[PS+2*(Q17-1)+2]=1
THEN
PSB:=PFT3[Q17,K16];
IF A[Q17]=1 AND A[PS+2*(Q17-1)+1]=0 AND A[PS+2*(Q17-1)+2]=0
THEN
PSB:=PFT4[Q17,K16];
IF A[Q17]=0 AND A[PS+2*(Q17-1)+1]=1 AND A[PS+2*(Q17-1)+2]=1
THEN
PSB:=PFT5[Q17,K16];
IF A[Q17]=1 AND A[PS+2*(Q17-1)+1]=1 AND A[PS+2*(Q17-1)+2]=0
THEN
PSB:=PFT6[Q17,K16];
IF A[Q17]=1 AND A[PS+2*(Q17-1)+1]=0 AND A[PS+2*(Q17-1)+2]=1
THEN
PSB:=PFT7[Q17,K16];
IF A[Q17]=1 AND A[PS+2*(Q17-1)+1]=1 AND A[PS+2*(Q17-1)+2]=1
THEN
PSB:=PFT8[Q17,K16];
END OF PSB;

GLTID:=TIME(2);                                                 6
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL N DO FH[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL N DO RH[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL S DO NRSB[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL M DO FS[Q]);
READ(INP620,/, FOR Q:=1 STEP 1 UNTIL N DO LNY[Q]);
READ(INP620,/, FOR Q:=1 STEP 1 UNTIL M DO FNY[Q]);
READ(INP620,/,          H0,EPS,T,F,MAXTID,X);
XORIG:=X;
HORIG:=H0;
FOR Q:=1 STEP 1 UNTIL S DO
BEGIN
NOPR:=0;                                                        6
H:=1;
P12:
IF NRSB[Q]>N-S THEN
BEGIN
H:=H+1;                                                         7
Q:=NRSB[Q]-N+S;
GO TO P12;
END;
```

```
HNR[Q]:=H;
END;
FOR Q6:=1 STEP 1 UNTIL S DO                              (2)
IF HNR[Q6]=1 THEN
FOR Q7:=1 STEP 1 UNTIL ND DO
BEGIN
Q9:=N-S+Q6;
T9:=Q7*T/ND;
VFT1[Q6,Q7]:=VFK(1,Q9,T9);
VFT2[Q6,Q7]:=VFK(2,Q9,T9);
VFT3[Q6,Q7]:=VFK(3,Q9,T9);
VFT4[Q6,Q7]:=VFK(4,Q9,T9);
END;
FOR Q6:=1 STEP 1 UNTIL S DO
IF HNR[Q6]=2 THEN
FOR Q7:=1 STEP 1 UNTIL ND DO
BEGIN
Q9:=N-S+Q6;
T9:=Q7*T/ND;
VFT5 [Q6,Q7]:=VFK(5 ,Q9,T9);
VFT6 [Q6,Q7]:=VFK(6 ,Q9,T9);
VFT7 [Q6,Q7]:=VFK(7 ,Q9,T9);
VFT8 [Q6,Q7]:=VFK(8 ,Q9,T9);
VFT9 [Q6,Q7]:=VFK(9 ,Q9,T9);
VFT10[Q6,Q7]:=VFK(10,Q9,T9);
VFT11[Q6,Q7]:=VFK(11,Q9,T9);
VFT12[Q6,Q7]:=VFK(12,Q9,T9);
END;
FOR Q18:=1 STEP 1 UNTIL PS DO
FOR Q19:=1 STEP 1 UNTIL ND DO
BEGIN
T9:=Q19*T/ND;
PFT1[Q18,Q19]:=PFK(1,Q18,T9);
PFT2[Q18,Q19]:=PFK(2,Q18,T9);
PFT3[Q18,Q19]:=PFK(3,Q18,T9);
PFT4[Q18,Q19]:=PFK(4,Q18,T9);
PFT5[Q18,Q19]:=PFK(5,Q18,T9);
PFT6[Q18,Q19]:=PFK(6,Q18,T9);
PFT7[Q18,Q19]:=PFK(7,Q18,T9);
PFT8[Q18,Q19]:=PFK(8,Q18,T9);
END;
WRITE(OUTP620,<"TID=",F10.2>,(TIME(2)-GLTID)/60);        (3)
E:=0;                                                    (4)
D:=0;
E2:=0;
D2:=0;
TAELLER:=0;
SIGMA1R:=0;
SIGMA1G:=0;
SIGMA2R:=0;
SIGMA2G:=0;
P15:
AF:=0;                                                   (5)
FOR Q:=N+1 STEP 1 UNTIL N+N DO A[Q]:=0;                  (6)
FOR Q:=1 STEP 1 UNTIL N DO
IF RANDOM(X) GEQ (1-FNY[Q]) THEN
A[Q+N]:=1;
FOR Q:=1 STEP 1 UNTIL 2*N DO HM2[Q]:=2*T;
Y:=0;                                                    (7)
FOR Q:=PS+1 STEP 1 UNTIL N-S DO                          (8)
HM1[Q]:=-LN(RANDOM(X))/LNY[Q];
FOR Q:=PS+1 STEP 1 UNTIL N-S DO
HM1[Q+N]:=HM1[Q]-LN(RANDOM(X))/RH[Q];
FOR Q:=1 STEP 1 UNTIL PS DO
BEGIN
NUMMER1:=PS+2*(Q-1)+1;
NUMMER2:=PS+2*(Q-1)+2;
```



```
IF HM1[NUMMER1]<HM1[NUMMER2]
THEN
MINNR:=NUMMER1
ELSE MINNR:=NUMMER2;
HM1[Q]:=HM1[MINNR] -LN(RANDOM(X))/LNY[Q];
HM1[Q+N]:=HM1[Q]-LN(RANDOM(X))/RH[Q];
END;
FOR Q:=1 STEP 1 UNTIL S DO
BEGIN
HM1[N-S+Q]:=HM1[NRSB[Q]]-LN(RANDOM(X))/LNY[Q];
HM1[2*N-S+Q]:=HM1[N-S+Q]-LN(RANDOM(X))/RH[N-S+Q];
END;
P8:  V:=HM1[N+1];                                        (9)
QMIN:=N+1;
FOR Q:=N+1 STEP 1 UNTIL 2*N DO
IF HM1[Q]<V THEN
BEGIN
V:=HM1[Q];
QMIN:=Q;
END;
KL:=0;
IF V>T THEN
BEGIN
V:=T;
KL:=1;
END;
G:=0;
TUR:=1;
IF QMIN LEQ N+3*PS THEN
GO TO P5;
IF QMIN LEQ(2*N-SBK) THEN
BEGIN
NR:=0;
GO TO P5;
END;
FOR H:=1 STEP 1 UNTIL S DO
IF QMIN=NRSB[H]+N
THEN BEGIN
HSB:=H;
G:=1;
END;
NR:=0;                                                   (10)
IF G=1 AND
HM1[N-S+HSB]> HM1[QMIN]
THEN NR:=1;
IF G=1 AND
HM1[N-S+HSB]>HM1[QMIN-N]
AND
HM1[N-S+HSB]<HM1[QMIN]
THEN NR:=2;
IF G=1 AND
HM1[N-S+HSB]<HM1[QMIN-N]
THEN NR:=3;
IF G=0 AND
HM1[NRSB[QMIN-2*N+S]]<HM1[QMIN]
AND
HM1[NRSB[QMIN-2*N+S]]>HM1[QMIN-N]
THEN NR:=4;
IF G=0 AND
HM1[QMIN-N]>HM1[NRSB[QMIN-2*N+S]]
THEN NR:=5;
IF G=0
AND
HM1[NRSB[QMIN-2*N+S]]>HM1[QMIN]
THEN NR:=6;
P5:
FOR P:=1 STEP 1 UNTIL N DO
A[P]:=0;
```

```
IF HM1[H]<V
OR(HM2[H]<V AND HM2[H+N]>V)THEN
A[H]:=1;
B:=0;
IF                                          (11)
{A[1]=1 AND (A[2]=1 OR A[3]=1)}  INPUT: System failure conditions   (12)
                                            (13)
THEN BEGIN                                  (14)
B:=1;
AF:=AF+1;
END;
IF B=1 THEN Y:=1;
IF B=1 AND(AF=1 OR KL=1) THEN               (15)
BEGIN
VAEGTETR:=1;
Q8:=ENTIER(V*ND/T+0.5);
IF Q8=0 THEN Q8:=1;
FOR Q:=1 STEP 1 UNTIL PS DO
VAEGTETR:=VAEGTETR*PSB(Q,Q8);
FOR Q:=3*PS+1 STEP 1 UNTIL N-SBK DO
BEGIN
IF A[Q]=1 THEN
VAEGTETR:=VAEGTETR*(1-(RH[Q]+FH[Q]*EXP(-(FH[Q]+RH[Q])*V))/
( FH[Q]+RH[Q]))/
(1-(RH[Q]+LNY[Q]*EXP(-(LNY[Q]+RH[Q])*V))/
(LNY[Q]+RH[Q]));
IF A[Q]=0 THEN
VAEGTETR:=VAEGTETR*(RH[Q]+FH[Q]*EXP(-(FH[Q]+RH[Q])*V))/(FH[Q]+RH[Q])/
(RH[Q]+LNY[Q]*EXP(-(LNY[Q]+RH[Q])*V))*(LNY[Q]+RH[Q]);
END;
FOR P:=N-S+1 STEP 1 UNTIL N DO
BEGIN
Q3:=0;
FOR Q2:=1 STEP 1 UNTIL S DO
IF P= NRSB[Q2] THEN Q3:=1;
IF Q3=0 THEN
VAEGTETR:=VAEGTETR*VSB(P,Q8);
END;
FOR Q:=1 STEP 1 UNTIL M DO
BEGIN
IF A[Q+N]=1 THEN
VAEGTETR:=VAEGTETR*FS[Q]/FNY[Q];
IF A[Q+N]=0 THEN
VAEGTETR:=VAEGTETR*(1-FS[Q])/(1-FNY[Q]);
END;
END;
P7:
IF AF=1 THEN VB1:=VAEGTETR;                 (16)
IF KL=1 THEN GO TO P6;                      (17)
FOR Q:=1 STEP 1 UNTIL 2*N DO                (18)
HM2[Q]:=HM1[Q];
IF QMIN LEQ N+3*PS THEN                     (19)
BEGIN
FOR Q:=1 STEP 1 UNTIL PS DO
BEGIN
IF Q = QMIN-N
THEN
BEGIN
NUMMER1:=PS+2*(Q-1)+1;
NUMMER2:=PS+2*(Q-1)+2;
END;
IF PS+2*(Q-1)+1=QMIN-N THEN
BEGIN
NUMMER1:=Q;
NUMMER2:=PS+2*(Q-1)+2;
END;
```



```
IF PS+2*(Q-1)+2=QMIN-N THEN
BEGIN
NUMMER1:=Q;
NUMMER2:=PS+2*(Q-1)+1;
END;
IF HM1[NUMMER1]<HM1[NUMMER2]
THEN MINNR:=NUMMER1
ELSE MINNR:=NUMMER2;
HM1[QMIN-N]:=HM1[MINNR]-LN(RANDOM(X))/LNY[QMIN-N];
HM1[QMIN]:=HM1[QMIN-N]-LN(RANDOM(X))/RH[QMIN-N];
END;
FOR Q:=1 STEP 1 UNTIL M DO
IF RANDOM(X) GEQ (1-FNY[Q]) THEN
A[Q+N]:=1;
GO TO P6;
END;
IF NR=0 OR NR=3 OR NR=4 OR NR=5 THEN
BEGIN
HM1[QMIN-N]:=HM1[QMIN]-LN(RANDOM(X))/LNY[QMIN-N];
HM1[QMIN]:=HM1[QMIN-N]-LN(RANDOM(X))/RH[QMIN-N];
END;
IF NR=1 THEN
BEGIN
HM1[QMIN-N]:=HM1[N-S+HSB]-LN(RANDOM(X))/LNY[QMIN-N];
HM1[QMIN]:=HM1[QMIN-N]-LN(RANDOM(X))/RH[QMIN-N];
END;
IF NR=2 THEN
BEGIN
HM1[QMIN-N]:=HM1[QMIN]-LN(RANDOM(X))/LNY[QMIN-N];
HM1[QMIN]:=HM1[QMIN-N]-LN(RANDOM(X))/RH[QMIN-N];
END;
IF NR=6 THEN
BEGIN
HM1[QMIN-N]:=HM1[NRSB[QMIN-2*N+S]]-LN(RANDOM(X))/LNY[QMIN-N];
HM1[QMIN]:=HM1[QMIN-N]-LN(RANDOM(X))/RH[QMIN-N];
END;
FOR Q:=N+1 STEP 1 UNTIL N+M DO A[Q]:=0;
FOR Q:=1 STEP 1 UNTIL M DO
IF RANDOM(X) GEQ (1-FNY[Q]) THEN
A[Q+N]:=1;
P6:
IF KL=0 THEN
GO TO  P8;
IF B=1                                      (20)
THEN D:=D+VAEGTETR;
IF Y=1 THEN E:=E+VB1;                       (21)
IF Y=1 THEN
E2:=E2+1;
IF B=1 THEN                                 (22)
D2:=D2+1;
TAELLER:=TAELLER+1;                         (23)
SIGMA1R:=SIGMA1R+(VB1*Y)**2;                (24)
SIGMA1G:=SIGMA1G+(VAEGTETR*B)**2;
IF TAELLER<F AND                            (25)
MAXTID*60-TIME(2)/60 > 15 THEN
GO TO P15;
R:=(TAELLER-E)/TAELLER;                     (26)
G1:=(TAELLER-D)/TAELLER;
SIGMA2R:=(SQRT(SIGMA1R/TAELLER-(1-R)**2))/SQRT(TAELLER);   (27)
SIGMA2G:=(SQRT(SIGMA1G/TAELLER-(1-G1)**2))/SQRT(TAELLER);  (28)
WRITE(OUTP620,<"DIVERSE INPUT STOERRELSER">);              (29)
WRITE(OUTP620,<"N=",J8>,N);
WRITE(OUTP620,<"M=",J8>,M);
WRITE(OUTP620,<"S=",J8>,S);
WRITE(OUTP620,<"PS=",J8>,PS);
WRITE(OUTP620,<"ANTAL STANDBYKOBLEDE KOMPONENTER=",J8>,SBK);
WRITE(OUTP620,<"ANTAL TIDSSTEP FOR VAEGTFAKTOR=",J8>,ND);
WRITE(OUTP620,<"FEJLHYPPIGHEDER I TIMER**(-1)">);
```

```
FOR Q:=1 STEP 1 UNTIL N DO
WRITE(OUTP620,<"FH[",J5,"]=",E11.4>,Q,FH[Q]);
WRITE(OUTP620,<"REPARATIONSHYPPIGHEDER I TIMER**(-1)">);
FOR Q:=1 STEP 1 UNTIL N DO
WRITE(OUTP620,<"RH[",J5,"]=",E11.4>,Q,RH[Q]);
WRITE(OUTP620,<"DRIFTSKOMPONENT",X5,"STANDBYKOMPONENT">);
FOR Q:=1 STEP 1 UNTIL S DO
WRITE(OUTP620,<X4,I4,X17,I4>,NRSB[Q],N-S+Q);
WRITE(OUTP620,<"FEJLSANDSYNLIGHEDER">);
FOR Q:=1 STEP 1 UNTIL M DO
WRITE(OUTP620,<"FS[",J5,"]=",E11.4>,Q+N,FS[Q]);
WRITE(OUTP620,<"KORRIGEREDE FEJLHYPPIGHEDER:">);
FOR Q:=1 STEP 1 UNTIL N DO
WRITE(OUTP620,<"LNY[",J5"]=",E11.4>,Q,LNY[Q]);
WRITE(OUTP620,<"KORRIGEREDE FEJLSANDSYNLIGHEDER:">);
FOR Q:=1 STEP 1 UNTIL M DO
WRITE(OUTP620,<"FNY[",J5"]=",E11.4>,Q+N,FNY[Q]);
WRITE(OUTP620,<"INPUT TIL DIFFSYS:">);
WRITE(OUTP620,<"HO=",E11.4>,HORIG);
WRITE(OUTP620,<"EPS=",E11.4>,EPS);
WRITE(OUTP620,<"LL=",J8>,LL);
WRITE(OUTP620,<"T=",F12.2,"TIMER">,T);
WRITE(OUTP620,<"F=",J10>,F);
WRITE(OUTP620,<"MAX KOERETID I MINUTTER=",J8>,MAXTID);
WRITE(OUTP620,<"X=",J8>,XORIG);
WRITE(OUTP620,<"RESULTATER:">);
WRITE(OUTP620,<"ANTAL KOERTE MONTE CARLO FORSOEG=",J10>,TAELLER);
WRITE(OUTP620,<"FINGERET PAALIDELIGHED=",E11.4>,(TAELLER-E2)/TAELLER);
WRITE(OUTP620,<"FINGERET AVAILABILITY TIL TIDEN T=",E11.4>,
(TAELLER-D2)/TAELLER);
WRITE(OUTP620,<"PAALIDELIGHEDEN AF SYSTEMET ER LIG MED",F15.12>,R);
WRITE(OUTP620,<"STANDARDAFVIGELSE PAA PAALIDELIGHEDEN=",F15.12>,
SIGMA2R);
WRITE(OUTP620,<"AVAILABILITY TIL TIDSPUNKTET T ER LIG MED",F15.12>,G1);
WRITE(OUTP620,<"STANDARDAFVIGELSE PAA AVAILABILITY=",X6,F15.12>,
SIGMA2G);
WRITE(OUTP620,<"MED 95 PROCENT KONFIDENS ER SYSTEMETS FEJLSANDSYNLIGHED
">);
WRITE (OUTP620,<"LIG MED ",   E11.4,"PLUS MINUS",  E11.4>,1-R,2*
SIGMA2R);
END OF P620;
                                                                     5

                                                                     4
END;
                                                                     3
END;                                                                 2

$ INCLUDE "SA/141/SLUT"
 $POP LIST
END.
```


P 6 2 0 / V 4
= = = = = = =

```
BEGIN COMMENT AD44"PROGRAM NR 620/V4-RELY4;                              1

DEFINE                                                                   1
PROGRAMNO=620                                                      #,
HEADTEXT=<"PROGRAM NO 620/V4                                       ">#,
FIL5=INP620                                                        #,
FIL6=OUTP620                                                       #;
FILE OUTP620(KIND=70),INP620;


$ INCLUDE "SA/141/START"
 $POP LIST                                                               2
BEGIN
INTEGER N,M,F,X,TAELLER,Q,C,P,QMIN,H,L,Y,B,E,XORIG,S,G,HSB,D,SBK,TUR,NR,  3

KL,XNU,Q1;
REAL GLTID;
READ(INP620,/,N,M,S,SBK);
BEGIN
INTEGER ARRAY A[0:N+M],NRSB[0:S];                                        4

BEGIN
LABEL P1,P2,P3,P4,P5,P6,P7;                                              5

REAL T,V,U,R,G,VORIG,MAXTID;
REAL ARRAY MF,MR,KF,KR[0:N],FS[0:M],HM1[0:2*N], HM2[0:2*N];
REAL PROCEDURE RANDOM(X);
INTEGER X;
BEGIN
C:=X*125;                                                                6
X:=C MOD 2796203;
RANDOM:=X/2796203 END RANDOM;
REAL PROCEDURE LT(Q1);                                                   6
INTEGER Q1;
BEGIN
LT     :=((MF[Q1]+1)/KF[Q1]*(-LN(RANDOM(X))))**(1/(MF[Q1]+1));           6
END OF LT;
REAL PROCEDURE RT(Q1);                                                   6
INTEGER Q1;
BEGIN
RT     :=((MR[Q1]+1)/KR[Q1]*(-LN(RANDOM(X))))**(1/(MR[Q1]+1));           6
END OF RT;
GLTID:=TIME(2);                                                          6
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL N DO KF[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL N DO MF[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL N DO KR[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL N DO MR[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL S DO NRSB[Q]);
READ(INP620,/,FOR Q:=1 STEP 1 UNTIL M DO FS[Q]);
READ(INP620,/,T,F,MAXTID,X);
XORIG:=X;
E:=0;                                                                    2
D:=0;
TAELLER:=0;
P3:
FOR Q:=N+1 STEP 1 UNTIL N+M DO A[Q]:=0;                                  3
FOR Q:=1 STEP 1 UNTIL M DO
IF RANDOM(X) GEQ (1-FS[Q]) THEN
A[Q+N]:=1;
FOR Q:=1 STEP 1 UNTIL 2*N DO HM2[Q]:=2*T;
```

```
V:=0;
FOR Q:=1 STEP 1 UNTIL N-S DO                         (5)
HM1[Q]:=LT(Q);
FOR Q:=1 STEP 1 UNTIL N-S DO
HM1[Q+N]:=HM1[Q]+RT(Q);
FOR Q:=1 STEP 1 UNTIL S DO
BEGIN                                              6
HM1[N-S+Q]:=HM1[NRSB[Q]]+LT(N-S+Q);
HM1[2*N-S+Q]:=HM1[N-S+Q]+RT(N-S+Q);
END;
P1:  V:=HM1[N+1]; B:=0;                              (6)
QMIN:=N+1;
FOR Q:=N+1 STEP 1 UNTIL 2*N DO
IF HM1[Q]<V THEN
BEGIN                                              6
V:=HM1[Q];
QMIN:=Q;
END;                                               6
KL:=0;
IF V>T THEN
BEGIN                                              6
V:=T;
KL:=1;
END;                                               6
G:=0;
TUR:=1;
IF QMIN LEQ(2*N-SBK) THEN                            (7)
BEGIN                                              6
NR:=0;
GO TO P5;
END;                                               6
FOR H:=1 STEP 1 UNTIL S DO
IF QMIN=NRSB[H]+N
THEN BEGIN                                         6
HSB:=H;
G:=1;
END;                                               6
NR:=0;
IF G=1 AND
HM1[N-S+HSB]> HM1[QMIN]
THEN NR:=1;
IF G=1 AND
HM1[N-S+HSB]>HM1[QMIN-N]
AND
HM1[N-S+HSB]<HM1[QMIN]
THEN NR:=2;
IF G=1 AND
HM1[N-S+HSB]<HM1[QMIN-N]
THEN NR:=3;
IF G=0 AND
HM1[NRSB[QMIN-2*N+S]]<HM1[QMIN]
AND
HM1[NRSB[QMIN-2*N+S]]>HM1[QMIN-N]
THEN NR:=4;
IF G=0 AND
HM1[QMIN-N]>HM1[NRSB[QMIN-2*N+S]]
THEN NR:=5;
IF G=0
AND
HM1[NRSB[QMIN-2*N+S]]>HM1[QMIN]
THEN NR:=6;
IF NR=1 OR NR=2 THEN
BEGIN VORIG:=V;
V:=HM1[QMIN-N]+1,0⏨-8;                             6
END;                                               6
IF NR=5 THEN
BEGIN
VORIG:=V;                                          6
```



```
END;
IF NR=6 THEN
BEGIN VORIG:=V;                                    6
V:=HM1[QMIN-N]+1.0⏨-8;
END;                                               6
IF V > T+1.0⏨-9 THEN GO TO P7;                     6
P5:
FOR P:=1 STEP 1 UNTIL N DO
A[P]:=0;
FOR H:=1 STEP 1 UNTIL N DO
IF HM1[H]<V
OR(HM2[H]<V AND HM2[H+N]>V)THEN
A[H]:=1;
B:=0;                                                (8)
IF                                                   (9)
{A[1]=1 AND A[2]=1 }  INPUT: SYSTEM failure conditions   (10)
THEN B:=1;                                           (11)
IF B=1 THEN Y:=1;                                    (12)
P7:
IF(NR=1 OR NR=2 OR NR=5 OR NR=6) AND TUR=1
THEN
BEGIN
V:=VORIG;                                          6
TUR:=2;
GO TO P5;
END;
IF KL=1 THEN GO TO P6;                               (13) 6
FOR Q:=1 STEP 1 UNTIL 2*N DO
HM2[Q]:=HM1[Q];                                      (14)
IF NR=0 OR NR=3 OR NR=4 OR NR=5 THEN                 (15)
BEGIN
HM1[QMIN-N]:=HM1[QMIN]+LT(QMIN-N);                 6
HM1[QMIN]:=HM1[QMIN-N]+RT(QMIN-N);
END;
IF NR=1 THEN                                       6
BEGIN
HM1[QMIN-N]:=HM1[N-S+HSB]+LT(QMIN-N);              6
HM1[QMIN]:=HM1[QMIN-N]+RT(QMIN-N);
END;
IF NR=2 THEN                                       6
BEGIN
HM1[QMIN-N]:=HM1[QMIN]+LT(QMIN-N);                 6
HM1[QMIN]:=HM1[QMIN-N]+RT(QMIN-N);
FOR Q:=N+1 STEP 1 UNTIL N+M DO A[Q]:=0;
FOR Q:=1 STEP 1 UNTIL M DO
IF RANDOM(X) GEQ (1-FS[Q]) THEN
A[Q+N]:=1;
END;
IF NR=6 THEN                                       6
BEGIN
HM1[QMIN-N]:=HM1[NRSB[QMIN-2*N+S]]+LT(QMIN-N);     6
HM1[QMIN]:=HM1[QMIN-N]+RT(QMIN-N);
FOR Q:=N+1 STEP 1 UNTIL N+M DO A[Q]:=0;
FOR Q:=1 STEP 1 UNTIL M DO
IF RANDOM(X) GEQ (1-FS[Q]) THEN
A[Q+N]:=1;
END;
P6:                                                6
IF KL=0 THEN
GO TO P1;
IF B=1                                               16.
THEN D:=D+1;
IF Y=1 THEN E:=E+1;                                  17
TAELLER:=TAELLER+1;                                  (18)
IF TAELLER<F AND                                     19
```

```
MAXTID*60-TIME(2)/60 > 15 THEN
GO TO P3;                                                              (20)
R:=(TAELLER-E)/TAELLER;
G:=(TAELLER-D)/TAELLER;                                                (23)
WRITE(OUTP620,<"DIVERSE INPUT STOERRELSER">);
WRITE(OUTP620,<"N=",J8>,N);
WRITE(OUTP620,<"M=",J8>,M);
WRITE(OUTP620,<"S=",J8>,S);
WRITE(OUTP620,<"ANTAL STANDBYKOBLEDE KOMPONENTER=",J8>,SBK);
WRITE(OUTP620,<"WEIBULL PARAMETRE:">);
WRITE(OUTP620,<"KOMP.NR.",X4,"LEVETIDSFORDELING",X5,"REPARATIONSTIDSFORD
ELING">);
WRITE(OUTP620,<X14,"K",X11,"M",X12,"K",X11,"M">);
FOR Q:=1 STEP 1 UNTIL N DO
WRITE(OUTP620,<X3,J3,X4,E11.4,X1,E11.4,X2,E11.4,X1,E11.4>,Q,KF[Q],MF[Q],
KR[Q],MR[Q]);
WRITE(OUTP620,<"DRIFTSKOMPONENT",X5,"STANDBYKOMPONENT">);
FOR Q:=1 STEP 1 UNTIL S DO
WRITE(OUTP620,<X4,I4,X17,I4>,NRSB[Q],N-S+Q);
WRITE(OUTP620,<"FEJLSANDSYNLIGHEDER">);
FOR Q:=1 STEP 1 UNTIL M DO
WRITE(OUTP620,<"FS[",J5,"]=",E11.4>,Q+N,FS[Q]);
WRITE(OUTP620,<"T=",F12.2,"TIMER">,T);
WRITE(OUTP620,<"F=",J10>,F);
WRITE(OUTP620,<"MAX KOERETID I MINUTTER=",J8>,MAXTID);
WRITE(OUTP620,<"X=",J8>,XORIG);
WRITE(OUTP620,<"RESULTATER:">);
WRITE(OUTP620,<"ANTAL KOERTE MONTE CARLO FORSOEG=",J10>,TAELLER);
WRITE(OUTP620,<"PAALIDELIGHEDEN AF SYSTEMET ER LIG MED",F15.12>,R);
WRITE(OUTP620,<"STANDARDAFVIGELSE PAA PAALIDELIGHEDEN=",F15.12>,   (21)
SQRT(R*(1-R)/TAELLER));
WRITE(OUTP620,<"AVAILABILITY TIL TIDSPUNKTET T ER LIG MED",F15.12>,G);
WRITE(OUTP620,<"STANDARDAFVIGELSE PAA AVAILABILITY=",X6,F15.12>,   (22)
SQRT(G*(1-G)/TAELLER));
END OF P620;

WRITE(OUTP620,<"TID=",F10.2>,(TIME(2)-GLTID)/60);                      5
END;

END;                                                                    4

$ INCLUDE "SA/141/SLUT"                                                 3
 $POP LIST                                                              2
END.
```


3. INPUT AND OUTPUT

This chapter contains examples of input and the corresponding output for all four versions of the program.

The input consists of two parts. One is the conditions of failure for the system. This part of the input must be inserted between two specially marked blank cards in the program itself (see chapter 2, and it is always reproduced together with the output for control. In the present examples the failure conditions are shown in the corresponding listings of the program. In version 1, 2 and 4 the failure conditions correspond to a system with two components, one of which is a spare, having the system failure conditions that both components are failed, which must be expressed (see also the program listings) as follows: A (1) = 1 AND A (2) = 1. In version 3 the failure conditions correspond to a similar system but taking into account that the operator who carries out the switching between the units also has a probability of failure (= 0,4). In this case the failure conditions for the system are: A (1) = 1 AND (A (2) = 1 OR A (3) = 1).

The other part of the input are the data which are necessary to describe the analysis to be performed.

It is important that the numbering of the units is in accordance with the sequence which is shown on fig. 11.

All input data are format free. The output is shown from two calculations with version 3, one with approx. 1% accuracy and another with approx. 10% accuracy.

All constants used in the inputs and outputs are defined in the table on fig. 4.

Underneath each output is - for comparison - presented the results of calculations on the same problem by other methods.

INPUT, VERSION 1:

1) SYSTEM FAILURE CONDITIONS:
   A (1) = 1 AND A (2) = 1

2) DATA:

2,	Problem number
12, 6, 1972,	Date
2, 0, 1, 2, 2,	N, M, S, SBK, ND
1.202 E - 03, 1.805 E-03,	Failure rates, hours^{-1} for the N time dependent failures, in natural order, starting with unit no. 1.
0.1146, 0.1146	Repair rates, hours^{-1} for the time dependent failures for which the repair times are exponentially distributed in natural order starting with unit no. 1 (see also section A5). For units with gamma distributed repair times let the repair rate = 0.
0, 0,	Repair rates, hours^{-1} for the time dependent failures that are gamma distributed in natural order, starting with unit no. 1. For units with exponential repair times let the repair rate = 0.
1,	The numbers of the units that correspond to the standby units and listed in the same sequence.
0,	Failure probabilities for the M time independent failures (none in this problem).
5.0 E-03, 5.0 E-03,	Increased failure rates, hours^{-1} for the N time dependent failures in natural order starting with unit no. 1.
0,	Increased failure probabilities for the M time independent failures in natural order starting with unit no. N+1.
10, 1.0 E-08,	DIFFSYS constants HO and EPS (see ref. 3).
1000, 10000, 3, 100001,	T, F, MAXTID, X,

OUTPUT, VERSION 1, ENGLISH TRANSLATION:

AEK PROGRAM NO. 620   PROBLEM NO. 2 - 12/6-1972
PROGRAM NO. 620/V1
TIME = 168.87
Various input data:
N = 2
M = 0
S = 1
Number of standby-coupled components (SBK) = 2
Number of time steps (ND) for the weighting factor = 5
Failure rate in hours^{-1}
FH (1) = 1.2020E-03
FH (2) = 1.8050E-03
Repair rates in hours^{-1}
A) With exponential distribution
   YX (1) = 1.1460E-01
   YX (2) = 1.1460E-01
B) With gamma distribution
   YG (1) = 0.0000E 00
   YG (2) = 0.0000E 00

Operational unit	Standby unit
1	2

Failure probabilities (for time independent failures, none in this problem).
Corrected failure rates
LNY (1) = 3.0000E-03
LNY (2) = 3.0000E-03
Corrected failure probabilities:

Input for DIFFSYS:
HO = 1.0000E 01
EPS= 1.0000E-08
LL = 3810

T = 1000 hours
F = 10000 (trials)

Max. computer time in minutes = 6

X = 100001.

Results:

Number of Monte Carlo simulations, actually run: 10000

Simulated reliability = 9.2980E-01

Simulated availability at time T = 9.9980E-01

The reliability of the system is equal to 0.982721092982

Standard deviation of the reliability = 0.000628843364

Availability at time T is equal to 0.999950772344

Standard deviation of the availability = 0.000034805728

With 95 percent confidence the failure probability of the system is equal to 1.7279E-02 plus or minus 1.2577E-03.

Result of a control calculation

By a Markov analysis the failure probability of the system was found to be: $1.81 \cdot 10^{-2}$.

-1-

```
A.E.K. PROGRAM NO.620      PROBLEM NO.2-  12/6.1972
PROGRAM NO 620/V1
TID=    168.87
DIVERSE INPUT STOERRELSER
N=2
M=0
S=1
ANTAL STANDBYKOBLEDE KOMPONENTER=2
ANTAL TIDSSTEP FOR VAEGTFAKTOR=5
FEJLHYPPIGHEDER I TIMER**(-1)
FH[1]= 1.2020E-03
FH[2]= 1.8050E-03
REPARATIONSHYPPIGHEDER I TIMER**(-1)
A)  MED EKSPONENTIEL FORDELING:
YX[1]= 1.1460E-01
YX[2]= 1.1460E-01
B)  MED GAMMA FORDELING:
YG[1]= 0.0000E 00
YG[2]= 0.0000E 00
DRIFTSKOMPONENT      STANDBYKOMPONENT
      1                      2
FEJLSANDSYNLIGHEDER
KORRIGEREDE FEJLHYPPIGHEDER:
LNY[1]= 3.0000E-03
LNY[2]= 3.0000E-03
KORRIGEREDE FEJLSANDSYNLIGHEDER:
INPUT TIL DIFFSYS:
H0= 1.0000E 01
EPS= 1.0000E-08
LL=3810
T=      1000.00TIMER
F=10000
MAX KOERETID I MINUTTER=6
X=100001
RESULTATER:
ANTAL KOERTE MONTE CARLO FORSOEG=10000
FINGERET PAALIDELIGHED= 9.2980E-01
FINGERET AVAILABILITY TIL TIDEN T= 9.9980E-01
PAALIDELIGHEDEN AF SYSTEMET ER LIG MED 0.982721092982
STANDARDAFVIGELSE PAA PAALIDELIGHEDEN= 0.000628843364
AVAILABILITY TIL TIDSPUNKTET T ER LIG MED 0.999950772344
STANDARDAFVIGELSE PAA AVAILABILITY=      0.000034805728
MED 95 PROCENT KONFIDENS ER SYSTEMETS FEJLSANDSYNLIGHED
LIG MED  1.7279E-02PLUS MINUS 1.2577E-03
```

PROCES TIME

INPUT, VERSION 2:

1) SYSTEM FAILURE CONDITIONS:
   A (1) = 1 AND A (2) = 1
2) DATA:

15,	Problem number.
9, 6, 1972	Date.
2, 0, 1, 2,	N, M, S, SBK,
1.202E-03, 1.805E-03	Failure rates, hours^{-1} for the N time dependent failures in natural order, starting with unit no. 1.
1.146E-01, 1.146E-01,	Repair rates, hours^{-1}, for the N time dependent failures in natural order, starting with unit no. 1. (see also section A5).
1,	The numbers of the units that correspond to the standby units and listed in the same sequence.
0,	Failure probabilities for the M time independent failures.
1000, 10000, 1, 100001,	T, F, MAXTID, X,

OUTPUT, VERSION 2, ENGLISH TRANSLATION:

AEK PROGRAM NO. 620 PROBLEM NO. 15 - 9/6-1972

PROGRAM NO. 620/V2

Various input data:

N = 2

M = 0

S = 1

Number of standby-coupled components = 2

Failure rates in hours^{-1}

FH (1) = 1.2020E-03

FH (2) = 1.8050E-03

Repair rates in hours^{-1}

RH (1) = 1.1460E-01

RH (2) = 1.1460E-01

Operational unit	Standby unit
1	2

Failure probabilities (for time independent failures, none in this problem)

T = 1000 hours

F = 10000

Max. computer time in minutes = 1

X = 100001

Results:

Number of Monte Carlo trials, actually run	= 8840
The reliability of the system is equal to	0.983936651584
Standard deviation of the reliability	= 0.001337135676
Availability at time T is equal to	0.999886877828
Standard deviation of availability	= 0.000113115774
Time	= 44.89

Result of a control calculation:

By a Markov analysis the failure probability of the system was found to be: $1.81 \cdot 10^{-2}$.

OUTPUT, VERSION 2, ORIGINAL TEXT

```
A.E.K. PROGRAM NO.620      PROBLEM NO.15-    9/6.1972
PROGRAM NO 620/V2
DIVERSE INPUT STOERRELSER
N=2
M=0
S=1
ANTAL STANDBYKOBLEDE KOMPONENTER=2
FEJLHYPPIGHEDER I TIMER**(-1)
FH[1]= 1.2020E-03
FH[2]= 1.8050E-03
REPARATIONSHYPPIGHEDER I TIMER**(-1)
RH[1]= 1.1460E-01
RH[2]= 1.1460E-01
DRIFTSKOMPONENT      STANDBYKOMPONENT
      1                      2
FEJLSANDSYNLIGHEDER
T=      1000.00TIMER
F=10000
MAX KOERETID I MINUTTER=1
X=100001
RESULTATER:
ANTAL KOERTE MONTE CARLO FORSOEG=6840
PAALIDELIGHEDEN AF SYSTEMET ER LIG MED 0.983936651584
STANDARDAFVIGELSE PAA PAALIDELIGHEDEN= 0.001337135676
AVAILABILITY TIL TIDSPUNKTET T ER LIG MED 0.999886877828
STANDARDAFVIGELSE PAA AVAILABILITY=        0.000113115774
TID=      44.89
```

PROCES TIME = 1/5 SEC.

INPUT, VERSION 3:

1) SYSTEM FAILURE CONDITIONS:

A (1) = 1 AND (A (2) = 1 OR A (3) = 1)

2) DATA:

27,	Problem number,
9, 6, 1972	Date,
2, 1, 1, 0, 2, 5	N, M, S, PS, SBK, ND,
4.3000E-09, 4.3000E-09,	Failure rates, hours^{-1} for the N time dependent failures in natural order, starting with unit no. 1.
1.0E-10, 1.0E-10, (no repair)	Repair rates, hours^{-1} for the N time dependent failures in natural order, starting with unit no. 1 (see also section A5).
1,	The numbers of the units that correspond to the standby units and listed in the same sequence.
0.4,	Failure probabilities for the M time independent failures.
2.3E-04, 2.3E-04,	Increased failure rates, hours^{-1} for the N time dependent failures in natural order, starting with unit no. 1.
0.4,	Increased (no increase in this case) failure probabilities for the M time independent failures in natural order, starting with unit no. N+1.
10, 1.0E-08,	DIFFSYS constants HO and EPS (see ref. 3).

1000, 1000000, 15, 100001, T, F, MAXTID, X,

OUTPUT, VERSION 3, ENGLISH TRANSLATION:

AEK PROGRAM NO. 620 PROBLEM NO. 27 - 9/6-1972

PROGRAM NO. 620/V3

Time = 12.78

Various input data

N = 2

M = 1

S = 1

PS= 0

Number of standby-coupled components (SBK) = 2

Number of time steps for weighting factor = 5

Failure rates in hours^{-1}

FH (1) = 4.3000E-09

FH (2) = 4.3000E-09

Repair rates in hours^{-1}

RH (1) = 1.0000E-10

RH (2) = 1.0000E-10

Operational unit	Standby unit
1	2

Failure probabilities

FS (3) = 4.0000E-01

Corrected failure rates:

LNY (1)= 2.3000E-04

LNY (2)= 2.3000E-04

Corrected failure probabilities:

FNY (3)= 4.0000E-01

Input for DIFFSYS:

HO = 1.0000E 01

EPS= 1.0000E-08

LL = 210

T = 1000 hours

F = 1000000

Max. computer time in minutes = 15

X = 100001

Results:

Number of Monte Carlo trials, actually run = 319097

Simulated reliability = 9.0399E-01

Simulated availability at time T = 9.0399E-01

The reliability of the system is equal to 0.999998280597

Standard deviation of the reliability = 0.000000010841

Availability at time T is equal to 0.999998280597

Standard deviation of the availability = 0.000000010841

With 95% confidence the failure probability of the system is equal to 1.7194E-06 plus or minus 2.1682E-08.

Result of a control calculation:

By a compound-event analysis the failure probability of the system was found to be: $1.72 \cdot 10^{-6}$.

OUTPUT, VERSION 3, ORIGINAL TEXT; HIGH ACCURACY


```
A.E.K. PROGRAM NO.620      PROBLEM NO.27-    9/6.1972
PROGRAM NO 620/V3
TID=     12.78
DIVERSE INPUT STOERRELSER
N=2
M=1
S=1
PS=0
ANTAL STANDBYKOBLEDE KOMPONENTER=2
ANTAL TIDSSTEP FOR VAEGTFAKTOR=5
FEJLHYPPIGHEDER I TIMER**(-1)
FH[1]= 4.3000E-09
FH[2]= 4.3000E-09
REPARATIONSHYPPIGHEDER I TIMER**(-1)
RH[1]= 1.0000E-10
RH[2]= 1.0000E-10
DRIFTSKOMPONENT      STANDBYKOMPONENT
      1                     2
FEJLSANDSYNLIGHEDER
FS[3]= 4.0000E-01
KORRIGEREDE FEJLHYPPIGHEDER:
LNY[1]= 2.3000E-04
LNY[2]= 2.3000E-04
KORRIGEREDE FEJLSANDSYNLIGHEDER:
FNY[3]= 4.0000E-01
INPUT TIL DIFFSYS:
HO= 1.0000E 01
EPS= 1.0000E-08
LL=210
T=      1000.00TIMER
F=1000000
MAX KOERETID I MINUTTER=15
X=100001
RESULTATER:
ANTAL KOERTE MONTE CARLO FORSOEG=319097
FINGERET PAALIDELIGHED= 9.0399E-01
FINGERET AVAILABILITY TIL TIDEN T= 9.0399E-01
PAALIDELIGHEDEN AF SYSTEMET ER LIG MED 0.999998280597
STANDARDAFVIGELSE PAA PAALIDELIGHEDEN= 0.000000010841
AVAILABILITY TIL TIDSPUNKTET T ER LIG MED 0.999998280597
STANDARDAFVIGELSE PAA AVAILABILITY=         0.000000010841
MED 95 PROCENT KONFIDENS ER SYSTEMETS FEJLSANDSYNLIGHED
LIG MED  1.7194E-06PLUS MINUS 2.1682E-08
```

PROCES TIME = 885 sec.

OUTPUT, VERSION 3, ORIGINAL TEXT; NORMAL ACCURACY


```
A.E.K. PROGRAM NO.620      PROBLEM NO.27-    8/6.1972
PROGRAM NO 620/V3
TID=     12.61
DIVERSE INPUT STOERRELSER
N=2
M=1
S=1
PS=0
ANTAL STANDBYKOBLEDE KOMPONENTER=2
ANTAL TIDSSTEP FOR VAEGTFAKTOR=5
FEJLHYPPIGHEDER I TIMER**(-1)
FH[1]= 4.3000E-09
FH[2]= 4.3000E-09
REPARATIONSHYPPIGHEDER I TIMER**(-1)
RH[1]= 1.0000E-10
RH[2]= 1.0000E-10
DRIFTSKOMPONENT      STANDBYKOMPONENT
      1                     2
FEJLSANDSYNLIGHEDER
FS[3]= 4.0000E-01
KORRIGEREDE FEJLHYPPIGHEDER:
LNY[1]= 2.3000E-04
LNY[2]= 2.3000E-04
KORRIGEREDE FEJLSANDSYNLIGHEDER:
FNY[1]= 4.0000E-01
INPUT TIL DIFFSYS:
HO= 1.0000E 01
EPS= 1.0000E-08
LL=210
T=      1000.00TIMER
F=10000
MAX KOERETID I MINUTTER=1
X=100001
RESULTATER:
ANTAL KOERTE MONTE CARLO FORSOEG=10000
FINGERET PAALIDELIGHED= 9.0320E-01
FINGERET AVAILABILITY TIL TIDEN T= 9.0320E-01
PAALIDELIGHEDEN AF SYSTEMET ER LIG MED 0.999998296398
STANDARDAFVIGELSE PAA PAALIDELIGHEDEN= 0.000000060978
AVAILABILITY TIL TIDSPUNKTET T ER LIG MED 0.999998296398
STANDARDAFVIGELSE PAA AVAILABILITY=         0.000000060978
MED 95 PROCENT KONFIDENS ER SYSTEMETS FEJLSANDSYNLIGHED
LIG MED  1.7036E-06PLUS MINUS 1.2196E-07
```

PROCES TIME = 40 sec.

INPUT, VERSION 4:

1) SYSTEM FAILURE CONDITIONS:

A (1) = 1 AND A (2) = 1

2) DATA:

6,	Problem number,
7, 12, 1971	Date,
2, 0, 1, 2,	N, M, S, SBK,
1.202E-03, 1.805E-03,	Weibull constants KF for the lifetime distributions of the N time dependent failures in natural order.
0, 0,	Weibull constants MF for the lifetime distributions of the N time dependent failures in natural order.
1.146E-01, 1.146E-01,	Weibull constants KR for the repairtime distributions of the N time dependent failures in natural order.
0, 0,	Weibull constants MR for the repairtime distributions of the N time dependent failures in natural order.
1,	The numbers of the units that correspond to the standby units and listed in the same sequence.
0,	Failure probabilities for the M time independent failures,

1000, 10000, 4, 100001, T, F, MAXTID, X,

OUTPUT, VERSION 4, ENGLISH TRANSLATION:

AEK PROGRAM NO. 620 PROBLEM NO. 6 - 7/12-1971

PROGRAM NO. 620/V4

Various input data:

N = 2

M = 0

S = 1

Number of standby-coupled components = 2

Weibull parameters:

Unit no.	Lifetime-distribution K	Lifetime-distribution M	Repairtime-distribution K	Repairtime-distribution M
1	1.2020E-03	0.0000E 00	1.1460E-01	0.0000E 00
2	1.8050E-03	0.0000E 00	1.1460E-01	0.0000E 00

Operational unit	Standby unit
1	2

Failure probabilities (for time-independent failures, none in this problem)

T = 1000 hours

F = 10000 (trials)

Max. computer time in minutes = 4

X = 100001

Results:

Number of Monte Carlo trials, actually run = 10000

The reliability of the system is equal to 0.9840000000

Standard deviation of the reliability = 0.001254750971

Availability at the time T is equal to 0.999900000001

Standard deviation of the availability = 0.000099995000

Time = 75,12

Result of a control calculation

By a Markov analysis the failure probability of the system was found to be: $1.81 \cdot 10^{-2}$.

OUTPUT, VERSION 4, ORIGINAL TEXT -1-

```
A.E.K. PROGRAM NO.620      PROBLEM NO.6-   7/12.1971
PROGRAM NO 620/V4
DIVERSE INPUT STOERRELSER
N=2
M=0
S=1
ANTAL STANDBYKOBLEDE KOMPONENTER=2
WEIBULL PARAMETRE:
KOMP.NR.    LEVETIDSFORDELING       REPARATIONSTIDSFORDELING
                K           M          K          M
   1      1.2020E-03  0.0000E 00  1.1460E-01  0.0000E 00
   2      1.8050E-03  0.0000E 00  1.1460E-01  0.0000E 00
DRIFTSKOMPONENT     STANDBYKOMPONENT
      1                   2
FEJLSANDSYNLIGHEDER
T=     1000.00TIMER
F=10000
MAX KOERETID I MINUTTER=4
X=100001
RESULTATER:
ANTAL KOERTE MONTE CARLO FORSOEG=10000
PAALIDELIGHEDEN AF SYSTEMET ER LIG MED 0.984000000000
STANDARDAFVIGELSE PAA PAALIDELIGHEDEN= 0.001254750971
AVAILABILITY TIL TIDSPUNKTET T ER LIG MED 0.999900000001
STANDARDAFVIGELSE PAA AVAILABILITY=       0.000099995000
TIO=     75.12
```

PROCES TIME 75 sec.

4. EXPERIENCE

4.1 General

The RELY 4 has been checked several times on a variety of systems for which the reliability and availability can be calculated by other methods. In all cases the results of the RELY 4 calculations have proved to be correct within the statistical uncertainties (see also the tests in section 2.3).

The methods that have been used for the checking have varied from case to case. As for simple parallel or series coupled components without repair the analytical expression for the reliability of the system have been derived directly (see ref. 1). For checking of calculations on standby systems either a Markov process has been used (see section A 4.2) or a method called the Compound-events approach (see ref. 1). In some cases calculations on more complicated systems have been verified by analytical methods by the Electronics dept. (ref. 6).

From experience version 3 has proved to be the most generally applicable version and it gives the highest accuracy for a given computer time. Therefore it is also the most used and "debugged" of the four versions of RELY 4.

In the sections 4.2-4.5 below are listed some of the most significant observations, that are peculiar to the individual versions of the program.

4.2 Version 1

Version 1 is the slowest of the two versions of the program that use importance sampling, due to the more tedious calculation of weighting factors (see the tests in section 2.3).

4.3 Version 2

This version uses direct simulation which is a very time consuming method as can be seen from the test in section 2.3. Version 2 is therefore only applicable for systems with very few components and relatively high failure rates.

4.4 Version 3

This version gives the highest accuracy for a given computer time and problem. In order to obtain the highest accuracy the increased failure rates (input data) should be selected so that the simulated reliability or availability which ever is being studied is > 0,9. The experience has shown that by doing so it has not been possible to detect errors in the results beyond the statistical uncertainties in any of the tests, and the statistical errors have in many cases been 1% or less. Selection of the increased failure rates is kind of an art since it is also a question of obtaining a small statistical error of the result for a given number of trials, see section A 4.6. As an example of selection of increased failure rates see the test of version 3 in chapter 3.

4.5 Version 4

This version uses direct simulation and its use is therefore limited like version 2 as mentioned in section 4.3 above. Version 4 can also handle systems with standby's but only with one 100% standby-unit in each standby-group.

4.6 Future Work

Future work on improvement of the program could for instance include following: incorporation into version 1 of the ability to handle 50% standby units, a change of version 3 to accommodate other distribution functions for the times to failure and repairtimes and an automatic evaluation of the necessary number of time steps (ND) for the weighting factors in version 1 and 3.

APPENDIX

Basic concepts and theories

A1 Definitions of reliability and availability

In this report reliability and availability is defined as follows in accordance with normal praxis:

> The reliability of a system is defined as the probability of obtaining a specified function of the system during an entire period of a specified length.
>
> The availability of a system at a given time T, is defined as the probability of obtaining a specified function of the system at the moment T.

A2 Generation of random periods by the Monte Carlo technique

The Monte Carlo technique is a method by which a stochastic phenomenon is simulated by means of an analogous random proces which behaves as much like the actual one as possible.

The Monte Carlo simulation in RELY 4 starts with generation of pseudorandom numbers, x, between 0 and 1 with rectangular distribution. This is done by the real procedure RANDOM (X) which is described in ref. 2. The procedure works this way:

1. Let C = 125 · X;
2. Let X = C modulus 2796203, (= remainder from division of C by 2796203).
3. Let Random (X) = X/2796203;

X is an integer, which at the beginning of the program has been assigned a specified starting value. The procedure will cause the values X and Random (X) to attain a subsequent series of values in the intervals 0-2796203, and 0-1 respectively with a very nearby rectangular distribution function (see ref. 2).

Then the following substitution is made: $t = t(x)$;

It can be proved (see ref. 1 p. 74) that the distribution function of t will be h(t) given by

$$h(t) = \left|\frac{dx}{dt}\right| \cdot f(x) \qquad \text{(I)},$$

where

f(x) is the distribution function of x, which in this case equals 1 constantly.

By proper choice of the function f(x) the desired distribution function h(t) for t can be obtained at least in principle.

Following three distribution functions are used in RELY 4.

1. Exponential distribution. This distribution function is used for both the time to failure and repair in version 2 and 3. The times to failure and repair are calculated by using the substitution

$$t = t(x) = -\frac{\ln x}{L};$$

This corresponds to $X = e^{-L \cdot t}$; $\frac{dx}{dt} = -L \cdot e^{-L \cdot t}$,

and from eq. (I): $h(t) = \left|\frac{dx}{dt}\right| \cdot f(x) = -L \cdot e^{-L \cdot t}$;

This means that t is exponentially distributed with a failure/repair rate L.

2. Weibull distribution. The Weibull distribution is used for both times to failure and repair in version 4. The following substitution is used

$$t = t(x) = \left(\frac{M+1}{K} \cdot (-\ln x)\right)^{\frac{1}{M+1}}$$

This gives $X = e^{-\frac{K}{M+1} \cdot t^{M+1}}$,

which means that

$$h(t) = \left|\frac{dx}{dt}\right| \cdot 1 = K \cdot t^{M} \cdot e^{-\frac{K}{M+1} \cdot t^{M+1}};$$

This is a Weibull distribution function with the constants K and M. If one lets M = 0, then $h(t) = K \cdot e^{-k \cdot t}$, which means that version 4 can handle both Weibull and exponential distribution functions for the times to failure and repair.

The times to failure and repair are calculated by the real procedures LT and RT respectively.

3. Gamma distribution. The Gamma distribution is used for the times to repair in version 1, while the times to failure are exponentially distributed. However, version 1 is also able to use exponentially distributed times to repair for any of the components, if desired, see chapter 3. The times to repair are calculated as:

$$t = -\frac{\ln(x_1)}{R} - \frac{\ln(x_2)}{R} - \frac{\ln(x_3)}{R}, \quad \text{(II)}$$

where x_1, x_2 and x_3 are subsequent numbers of RANDOM (X), generated by the real procedure RANDOM (X). As seen from the formula II, t is a sum of three exponentially distributed times with the repair rate R. According to ref. 1 p. 54 t is then Gamma distributed with the distribution function

$$h(t) = \frac{R^3}{2} \cdot t^2 \cdot e^{-R \cdot t}.$$

A3 Direct Simulation

Direct simulation is a simulation method by which all failure rates, repair rates and failure probabilities have the original values.

The accuracy of the method increases with the number of trials, but the method has the disadvantage that the required computation time for a given accuracy of the results increases tremendously, when the failure rates decrease or the number of units increases.

Even for small systems with less than ten components the computation time easily becomes prohibitive.

Following expression for the standard deviation, σ, of the reliability, R, determined by the direct simulation Monte Carlo method is derived in ref. 1 p. 507-8:

$$\sigma = \sqrt{\frac{R(1-R)}{N}}, \text{ where N is the total number of trials.}$$

A4 Importance Sampling

The principal idea in *importance sampling* is to concentrate the distribution of simulated system failure probability to the area of most importance: near the correct value. By direct simulation the system failure probability is being approached as an average of very frequent 0's (in trials with no system failure) and very seldom 1's, (in trials with system failure) see ref. 9.

The approach used in this program corresponds to the one which is outlined in ref. 5 p. 293. The Monte Carlo simulation is made on the basis of increased failure rates with a subsequent correction of the failure probability of the system by multiplication with a weighting factor. This way a given number of Monte Carlo trials will cause a relatively high number of concentrated weighted estimates of the probability of system failure. Thus the reliability estimate will relatively soon achieve a small statistical variation.

Let $P^{OP}_{N,ORIG}$ and $P^{FAIL}_{N,ORIG}$ denote the probability that component no. N is in the operating and failed condition respectively at the time of observation based upon the original failure rates and $P^{OP}_{N,INCR.}$ and $P^{FAIL}_{N,INCR}$ the corresponding figures with increased failure rates.

The weighting factor, W, that corrects the observed system failure rate with increased failure rates, can be calculated from:

$$W = \prod_{\substack{\text{all failed}\\\text{units}}} \frac{P^{FAIL}_{N,ORIG}}{P^{FAIL}_{N,INCR}} \cdot \prod_{\substack{\text{all operat.}\\\text{units}}} \frac{P^{OP}_{N,ORIG}}{P^{OP}_{N,INCR}} = \prod_{\substack{\text{all f.}\\\text{units}}} W_f \cdot \prod_{\substack{\text{all operat.}\\\text{units}}} W_0 \;;$$

where $W_f = \dfrac{P^{FAIL}_{N,ORIG}}{P^{FAIL}_{N,INCR}}$ (I) and $W_0 = \dfrac{P^{OP}_{N,ORIG}}{P^{OP}_{N,INCR}}$ (II) are weighting factors for the observed number of failed units and observed number of non-failed units respectively.

By direct simulation the probability of system failure is found as

$$FS = \frac{N_F}{F}, \text{ where}$$

N_F is the number of trials in which the system has failed and F is the total number of trials.

By importance sampling FS is found as

$$FS = \frac{\sum w \cdot y}{F}$$, where w is the weighting factor above and

y = 1 if the system has failed and y = 0 if the system is not failed. In other words in the final account of "failed" trials *each of them counts as* w "failed" trials, w being the corresponding weighting factor as described above, in contradiction to direct simulation where each "failed" trial counts as 1 "failed" trial.

The weighting factors are calculated as described in section A4.1, A4.2 and A4.3 below for three different categories of standby groups.

The weighting factors for these groups are evaluated at a specified number (ND) of equidistant moments throughout the period of observation and are then stored in a matrix. After each Monte Carlo simulation the weighting factor corresponding to the actual situation and the moment which is nearest to the time of observation is picked from above mentioned matrix.

In case the increased probability of system failure is so low that multiple failures of the system during the observed period appear very seldom the correction is adequate.

A4.1 Weighting factor for Units without Standby

The weighting factors for the units to which no standby units are assigned are found from the equations I and II above by inserting following expression for the availability A(t), at the time t, of a single unit (see ref. 1 p. 338)

$$A(t) = \frac{Y}{Y+L} + \frac{L}{Y+L} \cdot e^{-(L+Y)\cdot t}; \qquad \text{III}$$

where Y and L are the repair and failure rate resp.; This gives:

$$w_f = \frac{1 - A(t)_{N,ORIG}}{1 - A(t)_{N,INCR}} \text{ and } w_0 = \frac{A(t)_{n,ORIG}}{A(t)_{N,INCR}}.$$

Further

$$w_f = \frac{1 - (\frac{Y_0}{Y_0+L_0} + \frac{L_0}{Y_0+L_0} \cdot e^{-(L_0+Y_0) \cdot t})}{1 - (\frac{Y_0}{Y_0+L_I} + \frac{L_I}{Y_0+L_I} \cdot e^{-(L_I+Y_0)\cdot t})} ;$$

and

$$w_0' = \frac{\frac{Y_0}{Y_0+L_0} + \frac{L_0}{Y_0+L_0} \cdot e^{-(L_0+Y_0)\cdot t}}{\frac{Y_0}{Y_0+L_I} + \frac{L_I}{Y_0+L_I} \cdot e^{-(L_I+Y_0)\cdot t}} ;$$

Notice:the figures for the repair rates are not increased. Indices 0 and I stand for original and increased respectively.

A4.2 Weighting Factor for Standby Units

The weighting factor for standby units and their respective "operational units" are found by analysis of socalled Markov processes. A Markov proces is a stochastic transition between the possible failed states of a system, assuming 1) that each transition occurs independently of all others, 2) that the probability of a transition from a state of n occurrencies (f. instance n failed units) to the state of n+1 occurrencies in time Δt is $L \cdot \Delta t$, 3) that the transition probability of two or more occurrencies in time Δt is negligible (see ref. 1).

A Markov proces can be described by a Markov graph showing the infinitesimal transitions between the various possible failure states of a system. Such graphs are valuable tools for calculation of the infinitesimal transition probabilities between all the possible states of a system, from which the differential equations governing the probabilities of being in each of the possible states as a function of time, can be deduced.

This program can be used for systems incorporating a specified number of standby groups of following three different kinds:

1. Each unit has one 100% capacity standby.
2. Each unit has two 100% capacity standby's.
3. Two out of three groups.

Fig. 5 through 8 shows the Markov graphs for the availability of each of the above mentioned categories of standby units and from these figures following differential equations are found for each of the three kinds of standby groups.

L and Y are the failure rate and repair rate, respectively and P(N) is the probability that the group is at state no. N at time t.

1. (see fig. 6)

$\dot{P}(1) = Y1 \cdot P(2) + Y2 \cdot P(3) - L1 \cdot P(1);$

$\dot{P}(2) = Y2 \cdot P(4) + L1 \cdot P(1) - (Y1+L2) \cdot P(2);$

$\dot{P}(3) = Y1 \cdot P(4) - Y2 \cdot P(3);$

$\dot{P}(4) = L2 \cdot P(2) - (Y1+Y2) \cdot P(4);$

2. (see fig. 7)

$\dot{P}(1) = Y1 \cdot P(2) + Y2 \cdot P(3) + Y3 \cdot P(4) - L1 \cdot P(1);$

$\dot{P}(2) = Y2 \cdot P(5) + Y3 \cdot P(6) + L1 \cdot P(1) - (L2+Y1) \cdot P(2);$

$\dot{P}(3) = Y1 \cdot P(5) - Y2 \cdot P(3) + Y3 \cdot P(7);$

$\dot{P}(4) = Y2 \cdot P(7) + Y1 \cdot P(6) - Y3 \cdot P(4);$

$\dot{P}(5) = Y3 \cdot P(8) + L2 \cdot P(2) - (L3+Y1+Y2) \cdot P(5);$

$\dot{P}(6) = Y2 \cdot P(8) - (Y3+Y1) \cdot P(6);$

$\dot{P}(7) = Y1 \cdot P(8) - (Y2+Y3) \cdot P(7);$

$\dot{P}(8) = L3 \cdot P(5) - (Y1+Y2+Y3) \cdot P(8);$

3. (see fig. 8)

$\dot{P}(1) = Y1 \cdot P(2) + Y2 \cdot P(3) + Y3 \cdot P(4) - (L1+L2) \cdot P(1);$

$\dot{P}(2) = L1 \cdot P(1) + Y2 \cdot P(5) + Y3 \cdot P(6) - (L2+L3+Y1) \cdot P(2);$

$\dot{P}(3) = L2 \cdot P(1) + Y1 \cdot P(5) + Y3 \cdot P(7) - (Y2 +L1+L3) \cdot P(3);$

$\dot{P}(4) = Y1 \cdot P(6) + Y2 \cdot P(7) - (Y3+L1+L2) \cdot P(4);$

$\dot{P}(5) = L2 \cdot P(2) + L1 \cdot P(3) + Y3 \cdot P(8) - (Y3+Y1+L2) \cdot P(6);$

$\dot{P}(6) = L3 \cdot P(2) + L1 \cdot P(4) + Y2 \cdot P(8) - (Y3+Y1+L2) \cdot P(6);$

$\dot{P}(7) = L2 \cdot P(4) + L3 \cdot P(3) + Y1 \cdot P(8) - (Y3+Y2+L1) \cdot P(7);$

$\dot{P}(8) = L3 \cdot P(5) + L2 \cdot P(6) + L1 \cdot P(7) - (Y1+Y2+Y3) \cdot P(8);$

By integration of these differential equations from 0 to t, with the starting conditions that $P(1) = 1$ and all other $P(N)$'s = 0 the probabilities $P(N)$ are found. In this program the integration is carried out by the procedure DIFFSYS (ref. 3).

The weighting factor for each group of units with a standby is then found as

$$w = \frac{P(N)_{ORIG}}{P(N)_{INCR}} \text{ , where}$$

$P(N)_{ORIG}$ and $P(N)_{INCR}$ are the probabilities computed by the method above for the actual situation, which means at the actual time of observation and for the actual situation number N for the component group under consideration. Indices ORIG and INCR mean that $P(N)$ is calculated on the basic of the original and increased failure rates respectively.

A4.3 Weighting Factor for Time Independent Failures

The weighting factor w_{ti} for the time independent failures is found by means of the following expression:

$$w_{ti} = \prod \frac{FS_{ORIG}}{FS_{INCR}} \cdot \prod \frac{1 - FS_{ORIG}}{1 - FS_{INCR}} ;$$ where FS is the failure probability.

The first multiplication is carried out for all time independent failures in the failed condition, the last summation for all time independent failures in the non-failed condition at the moment of observation.

A4.4 Weighting Factor for Units with Gamma- or exponentially Distributed Times to Repair (Version 1)

As seen from section A2 the gamma distributed times to repair in version 1 of the program are generated as a sum of three exponentially distributed times.

The weighting factors are found by a Markov model; the Markov graph is shcwn on fig. 9. As seen the gamma distributed time to repair is simulated by introducing two intermediate dummy

states 2 and 3 each with an exponentially distributed time to repair. The graph is made in such a way that it can incorporate both gamma and exponentially distributed times to repair. For gamma distributed repair times Y1 = 0, for exponential Y2 = 0.

The differential equations for the Markov proces are:

$$\dot{P}(1) = Y1 \cdot P(2) + Y2 \cdot P(4) - L1 \cdot P(1);$$

$$\dot{P}(2) = L1 \cdot P(1) - (Y1+Y2) \cdot P(2);$$

$$\dot{P}(3) = Y2\,(P(2) - P(3));$$

$$\dot{P}(4) = Y2\,(P(3) - P(4));$$

The differential equations are integrated from t = 0 till t = t, time of observation, applying the starting conditions: $P(1) = 1$, $P(2) = P(3) = P(4) = 0$;

The integration is made by the procedure DIFFSYS.

The weighting factor for each unit is then calculated as:

$$w = \prod_{\substack{\text{all operating} \\ \text{units}}} \frac{P(1)_{ORIG}}{P(1)_{INCR}} \cdot \prod_{\substack{\text{all failed} \\ \text{units}}} \frac{P(2)_{ORIG}}{P(2)_{INCR}}$$

A4.5 Weighting Factor for Units with a Single Standby Unit and Times to Repair with a Gamma- or Exponential Distribution (Program Version 1)

The weighting factors for standby groups in version 1 of the program are calculated by a Markov model in a similar way as the weighting factors for single components in section A4.4 above. The times to repair can have either a gamma- or exponential distribution function. Version 1 of the program can only handle systems, where the standby groups consist of one operational unit and one 100% capacity standby unit.

The Markov graph for the standby groups in version 1 is shown on fig. 10.

The ifferential equations, governing the Markov proces are:

$\dot{P}(1) = YG1 \cdot P(8) + Y1 \cdot P(2) + YG2 \cdot P(12) + Y2 \cdot P(3) - L1 \cdot P(1);$

$\dot{P}(2) = YG2 \cdot P(6) + Y2 \cdot P(4) + L1 \cdot P(1) - (L2+Y1+YG1) \cdot P(2);$

$\dot{P}(3) = YG1 \cdot P(10) + Y1 \cdot P(4) - (Y2+YG2) \cdot P(3);$

$\dot{P}(4) = L2 \cdot P(2) - (Y1+YG1+Y2+YG2) \cdot P(4);$

$\dot{P}(5) = YG2 \cdot (P(4) - P(5));$

$\dot{P}(6) = YG2 \cdot (P(5) - P(6));$

$\dot{P}(7) = YG1 \cdot (P(2) - P(7));$

$\dot{P}(8) = YG1 \cdot (P(7) - P(8));$

$\dot{P}(9) = YG1 \cdot (P(4) - P(9));$

$\dot{P}(10) = YG1 \cdot (P(9) - P(10));$

$\dot{P}(11) = YG2 \cdot (P(3) - P(11));$

$\dot{P}(12) = YG2 \cdot (P(11) - P(12));$

The differential equations are integrated from 0 to t, time of observation. The starting conditions are P(1) = 1, P(2) = P(3) = ... P(12) = 0; the integration is made by means of the procedure DIFFSYS.

The weighting factor for each standby group is calculated by the formula:

$$w = \frac{P(N)_{ORIG}}{P(N)_{INCR}}$$, where the symbols have the same meaning as in section A4.2 above.

A4.6 Standard Deviation of the Result in Version 1 and 3

The probability of system failure by the importance sampling method is as already described calculated as

$$FS = \frac{\sum\limits_{\text{all trials}} w \cdot y}{F}$$, where

w is the resulting weighting factor, y = 1 in all trials with system failure, y = 0 in trials with no system failure, and F is the total number of Monte Carlo trials.

The standard deviation of FS is given by:

$$\sigma^2 (FS) = \sigma^2 \left(\frac{\sum w \cdot y}{F}\right) = \frac{1}{F} \sigma^2 (w \cdot y); \quad (I),$$

since FS is the average value of w · y.

Further, according to ref. 8 (p. 58):

$$\sigma^2 (w \cdot y) = E((w \cdot y)^2) - (E(w \cdot y))^2 \cong \frac{\sum (w \cdot y)^2}{F} - (FS)^2; \quad (II)$$

E stands for the statistical designation "Expected value of".

Equation I and II give following expression, by which the standard deviation of FS has been calculated:

$$\sigma (FS) = \frac{1}{\sqrt{F}} \sqrt{\frac{\sum\limits_{\text{all trials}} (w \cdot y)^2}{F} - (FS)^2}.$$

A5 Calculation of the effective repair rate

The value, Y, to be used for the repair rate in the case of exponentially distributed repair times is found by following formula:

$$Y = \frac{1}{T}, \text{ where}$$

$T = t_r + t_d$, and

t_r is the average repair time, the time that elapses between the observation and repair of a failure and t_d is the average dead time, the average time between the occurance and observation of a failure, which is found by the formula:

$$t_d = \frac{\tau}{L(1-e^{-L\tau})} - \frac{1}{L} \quad (\cong \tfrac{1}{2}\tau \text{ for } L\tau \ll 1);$$

τ is the time interval between tests (see ref. 8 p. 33).

ACKNOWLEDGEMENT

The author wishes to thank Mr. Dan S. Nielsen and Mr. Bjørn Runge from the Electronics Department for valuable assistance with independent control of the results of the calculations, and with the proof reading of this report. The author is also grateful to Miss Kirsten Nielsen for typing the manuscript and for patience with the authors terrible handwriting.

REFERENCES

1. Probabilistic Reliability. M.L. Shooman. Mc Graw-Hill New York, 1968.

2. Random numbers. ALGOL procedure: random, SA 51. Reactor Physics Department, Maj 1969.

3. Solution of ordinary differential equations. B6700-ALGOL-procedure DIFFSYS. SA/107 Reactor Physics Department, August 1971.

4. Monte Carlo Methods. J. Hammersley and D. Handscomb. Methuen & Co., LTD, London 1967.

5. Principles of unified systems safety analysis. B.J. Garrick. Nuclear Engineering and Design. Vol. 13 no. 2, August 1970.

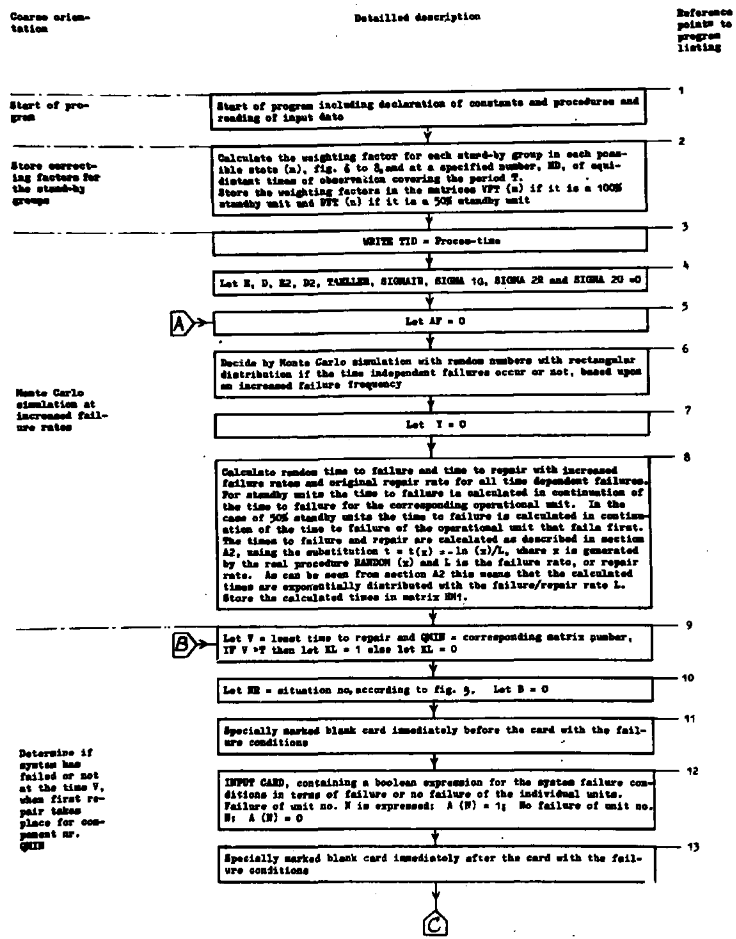
6. Pålidelighedsanalyse af Nødkølesystemet i Atomkraftværket Barsebäck. Risø-M-1470. H.E. Kongsø, Dan Nielsen og B. Runge. Februar 1972. FORTROLIG.

7. Reliability Analysis of Nuclear Power Plant Protective Systems. B.J. Garvick et al. HN-190. Holmes and Narver, Los Angelos, May 1967.

8. Safety and Reliability of Reactor Instrumentation with Redundant Instrument Channels. Risø-Report no. 34. Jens Rasmussen and P. Timmermann. January 1962.

9. Importance Sampling in Monte Carlo Analyses. Charles E. Clark. Operations Research, September-October 1961, p. 603 ff. (This is a very easy-to-understand explanation of the Monte Carlo technique and importance sampling.

Fig. 1. FLOW CHART FOR P620/VERSION 3



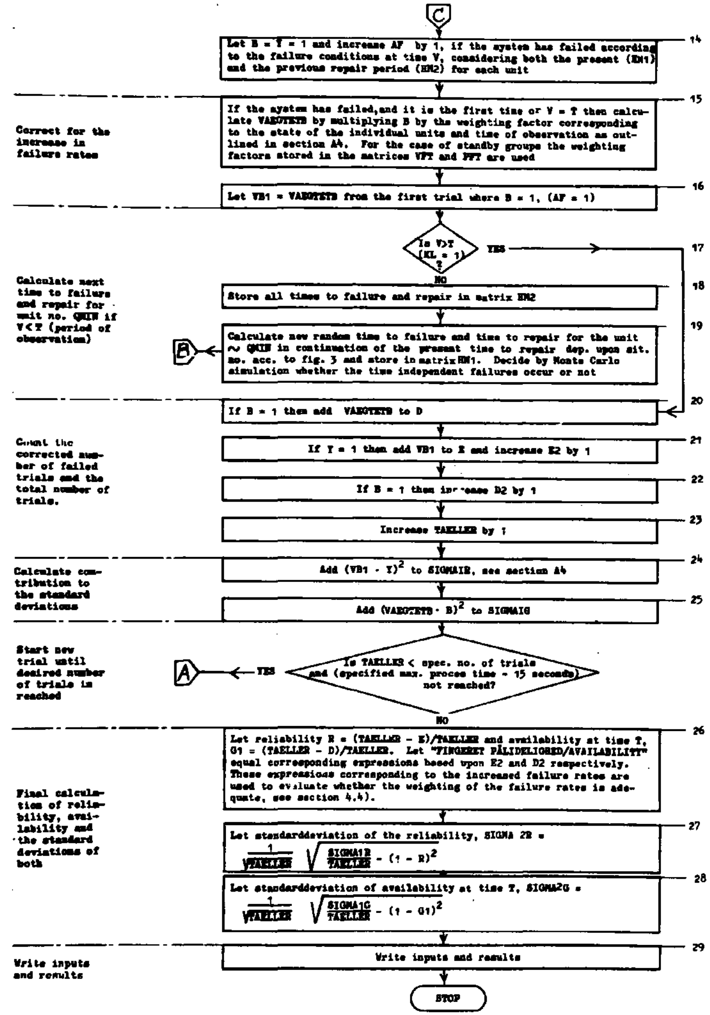
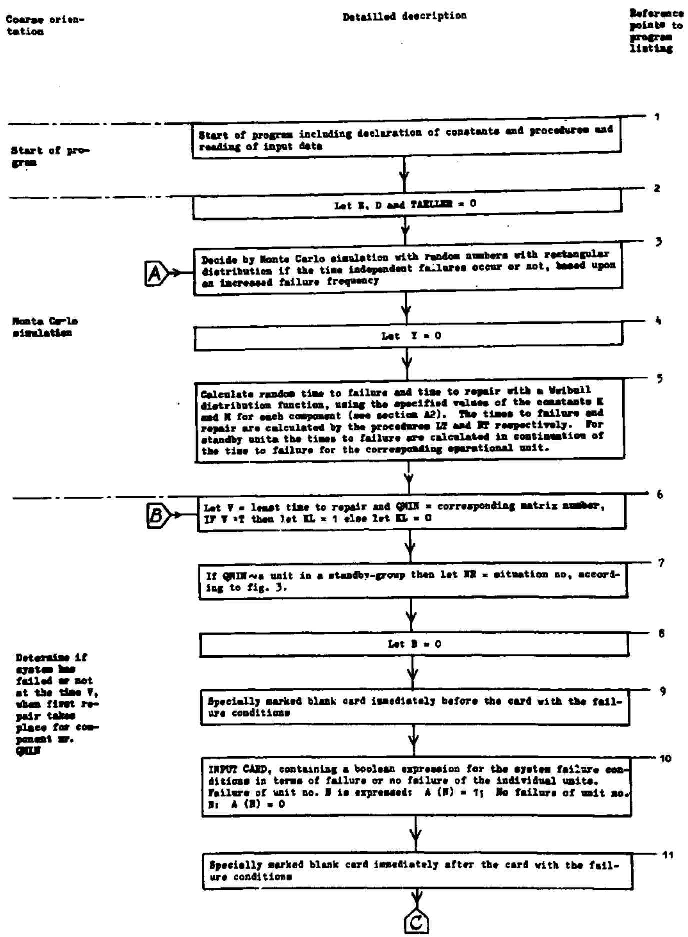
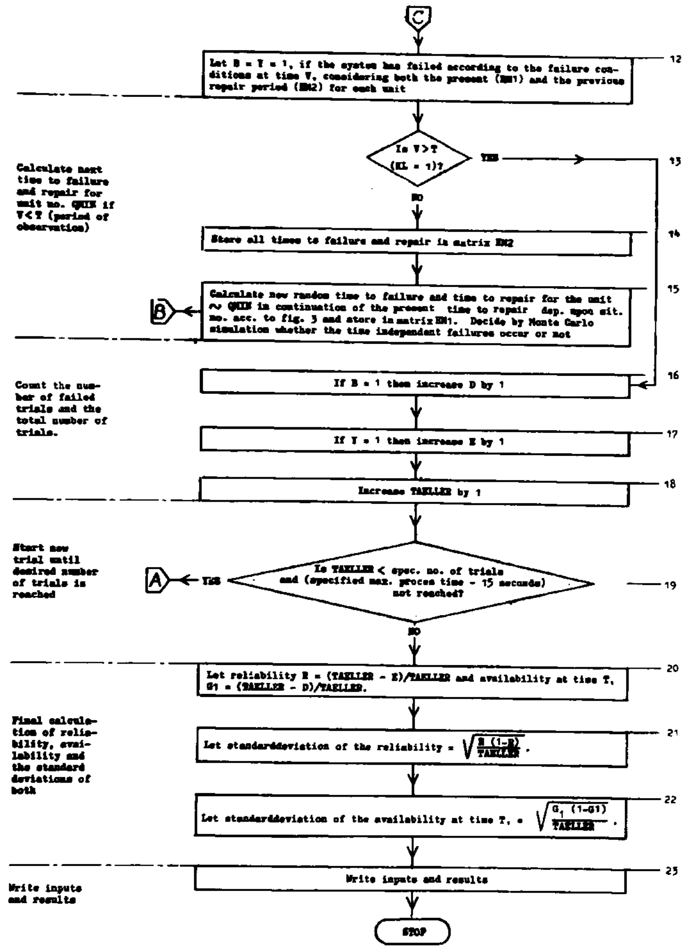
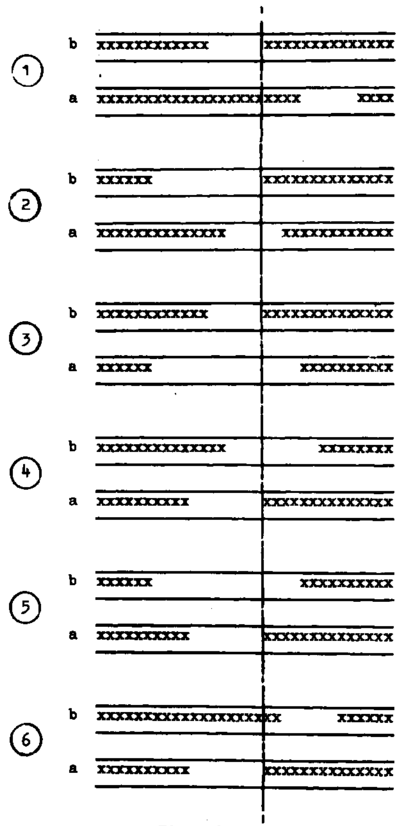


Fig. 2. FLOW CHART FOR P620/VERSION 4







Time of observation, V

Fig. 3. The possible relative locations of times to failure and times to repair for a standby unit a, and the corresponding operational unit b.

Signatures used: (n) : situation number

xxx: the unit is not failed

___: the unit is failed

NAME	UNIT	DEFINITION
AF		Number of system failures since latest start of a Monte Carlo simulation
A (n)		Integer array, = 1 if unit no n is failed and = 0 if unit no. n is not failed
B		B = 1 if the system is failed and B = 0 if the system is not failed at the moment of observation
D		Number of trials with system failures at time T, in version 1 and 3.D is corrected for the increase in the failure rates
D2		Number of trials with system failure at time T, based upon increased failure rates (in version 1 and 3)
E		Number of trials with system failure some time during period T, in version 1 and 3 corrected for the increase in the failure rates
E2		Number of trials with system failure some time during period T, based upon increased failure rates
EPS		DIFFSYS constant, see ref. 3
F		Desired number of Monte Carlo trials
FH (n)	$hours^{-1}$	Failure rate for unit no n
FNY (n)	$hours^{-1}$	Increased failure rate for unit no n
FS (n)		Failure probability for time independent failure no n
G1		Availability of the system at time T
HO		DIFFSYS constant, see ref. 3
HM1 (n)	hours	For $n \leq N$: HM1 (n)= Latest time to failure for component no n. For $n > N$: HM1 (n) = Latest time to repair for unit no n-N
HM2 (n)	hours	For $n \leq N$: HM2 (n) = Time to previous failure for component no. n. For $n > N$: HM2 (n) = time to previous repair for unit no. n-N.

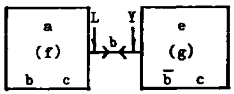
Fig. 4. Definition of constants from the program, that are being used in the block diagrams in chapter 2 or in the input and output examples in chapter 3

NAME	UNIT	DEFINITION
KF (n)		Weibull constant K for time to failure for unit no. n.
KL		If time of observation V ≦ T then KL = 0, else KL = 1
KR (n)		Weibull constant K for time to repair for unit no. n.
LL		Number of integration steps with the procedure DIFFSYS.
LNY (n)	hours^{-1}	Increased (corrected) failure rate for unit no. n.
MAXTID	min	Max. procestime. Shortly before this time the simulation will be stopped and the final calculations will be made.
MF (n)		Weibull constant M for the time to failure for unit no. n.
MR (n)		Weibull constant M for the time to repair for unit no. n.
N		Number of time dependent failures.
ND		Number of time steps for the weighting factor, see section A4.
NR		Situation number for standby units according to fig. 3.
PFT (n)		Matrix for storage of weighting factor for the 50% standby groups.
PS		Number of 50% standby units.
QMIN		Number of the unit with the smallest time to repair, plus N.
R		Reliability of the system.
RH (n)	hours^{-1}	Repair rate for unit no. n.
S		Number of 100% standby units.
SBK		Number of standby-coupled units (i.e. the standby units and the corresponding units).
SIGMA1G		$\text{SIGMA1G} = \sum_{\text{all trials}} (\text{VAEGTETB} \cdot \text{B})^2$
SIGMA1R		$\text{SIGMA1R} = \sum_{\text{all trials}} (\text{VB1} \cdot \text{Y})^2$

Fig. 4 (continued)

NAME	UNIT	DEFINITION
SIGMA2G		SIGMA2G is the standard deviation of the availability at time T.
SIGMA2R		SIGMA2R is the standard deviation of the reliability of the system.
T	hours	Period of observation = test interval for the entire system.
TAELLER		Number of Monte Carlo trials, that has actually been carried out.
V	hours	Time of observation = least time to repair.
VAEGTETB		The total weighting factor (see section A4).
VB1		The weighting factor corresponding to the first failure (AF = 1) of the system in each Monte Carlo trial.
VFT (n)		Matrix for storage of weighting factors for the 100% standby groups.
X		Starting number for X in the procedure RANDOM (X), see ref. 2.
Y		Y = 1 if the system has failed once or more during the entire period of observation else Y = 0.
YG (n)		YG (n) = $(\frac{1}{\beta})$ for gamma distributed time to repair for unit no. n (α = 2) in version 1.
YX (n)	hours^{-1}	Repair rate for unit no. n with exponential distribution in version 1.

Fig. 4 (continued)



a, e	:	state numbers
(f), (g)	:	state numbers, used in the program
b, c	:	unit numbers
bar above unit no.	:	unit in failed state
L	:	failure rate
Y	:	repair rate
letter on line (b)	:	number of the unit undergoing the transition

Fig.5 . Common signatures for the Markov graphs on fig. 6 through 10.

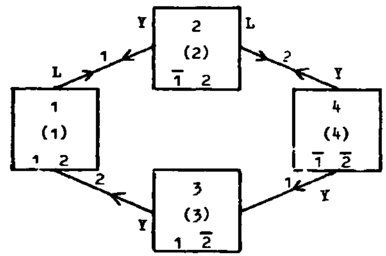


Fig. 6. Markov graph for the availability of a unit, 1, with a single standby, 2.

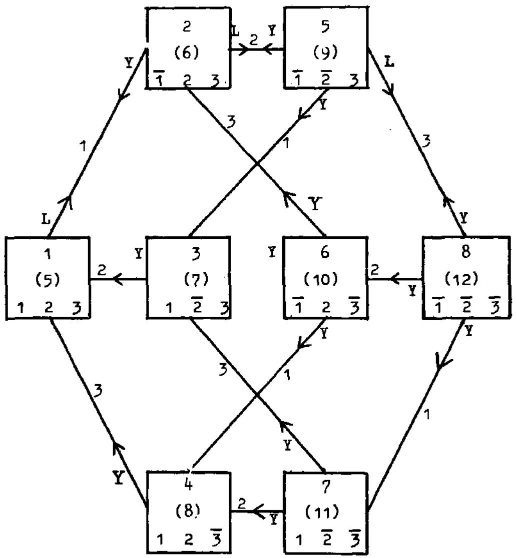


Fig. 7. Markov graph for the availability of a group consisting of one unit, 1, with two standby units, 2 and 3.

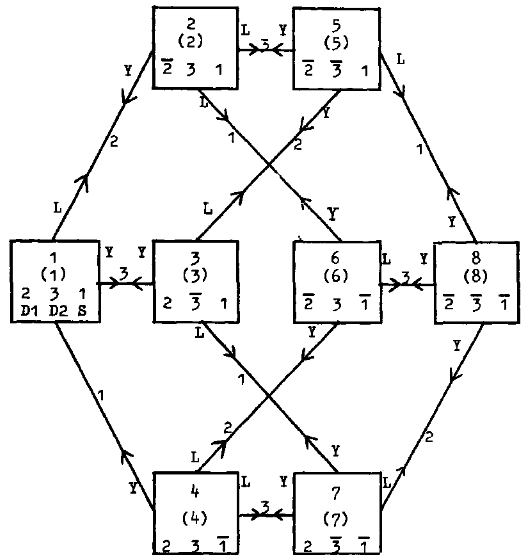


Fig. 8. Markov graph for the availability of a two-out-of-three group (unit 2 and 3 have both unit 1 as standby).

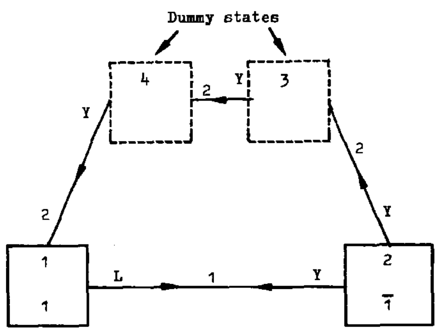


Fig. 9. Markov graph for the availability of a single unit with gamma ($Y1 = 0$, $Y2 \neq 0$) or exponentially ($Y1 \neq 0$, $Y2 = 0$) distributed time to repair.

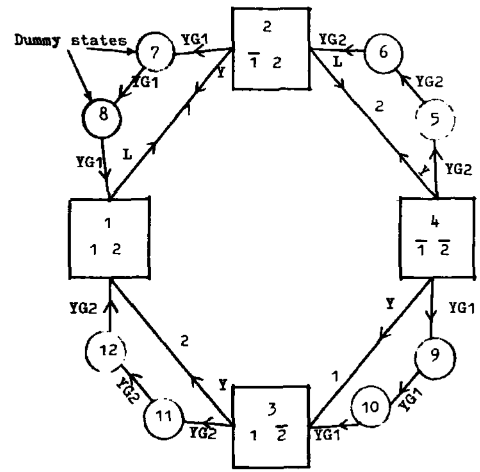
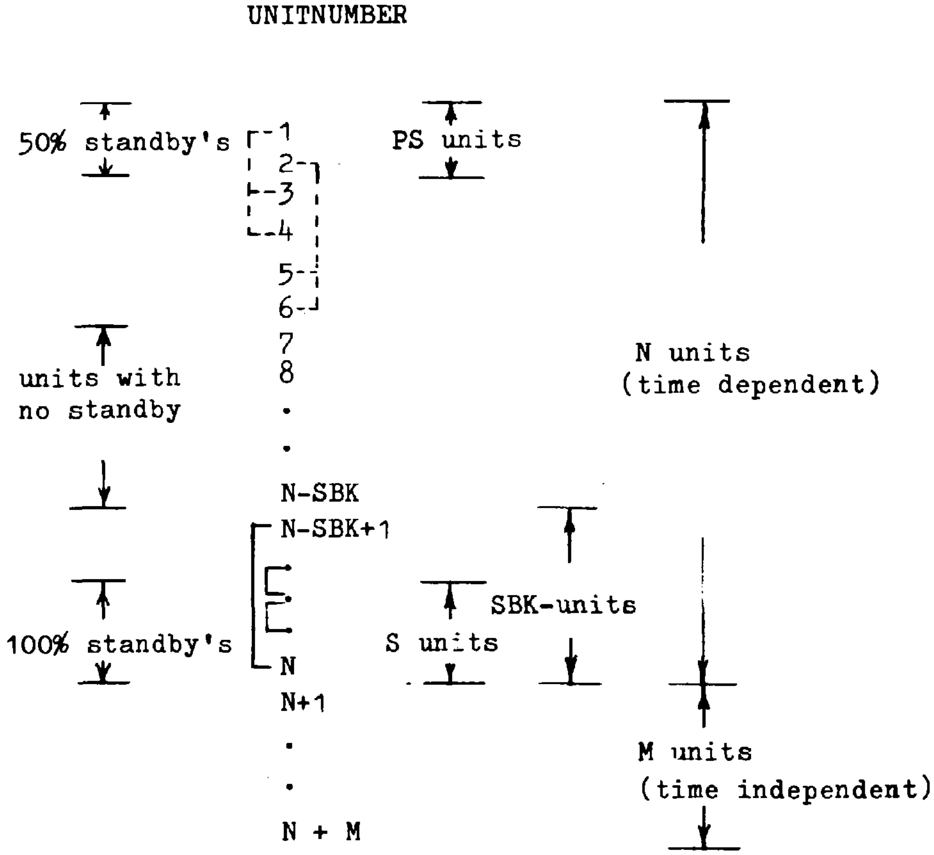


Fig.10. Markov graph for the availability of a unit (1) with a single 100% standby unit (2) with gamma ($YX = 0$, $YG \neq 0$) or exponential ($YX \neq 0$, $YG = 0$) distribution of the times to repair. Both units have exponentially distributed times to failure.



Signatures: ———————— 100% standby-coupling

---------- 50% standby-coupling

Fig. 11. Numbering of units